DTIC FILE COPY

AD-A202 600

D
E
S JAN 1 8 1989
H

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

89 1 17 150

①

AFIT/GE/ENG/88D-4

POSITION, SCALE, AND ROTATION INVARIANT OPTICAL PATTERN RECOGNITION FOR TARGET EXTRACTION AND IDENTIFICATION

THESIS

J. Thomas Walrond
Captain, USAF

Timothy G. Childress
Captain, USAF

AFIT/GE/ENG/88D-4

DTIC
S ELECTE D
JAN 1 8 1989
H

Approved for public release; distribution unlimited

AFIT/GE/ENG/88D-4

POSITION, SCALE, AND ROTATION INVARIANT OPTICAL PATTERN RECOGNITION FOR TARGET EXTRACTION AND IDENTIFICATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

J. Thomas Walrond, B.S.E.E.
Captain, USAF

Timothy G. Childress, B.S.E.E.
Captain, USAF

December, 1988

Approved for public release; distribution unlimited

Table of Contents

Page

Preface ii

Table of Contents iii

List of Figures vi

Abstract ix

I. Introduction 1
  1.1 Background 1
  1.2 Problem Definition 3
  1.3 Scope of Thesis 3
  1.4 Approach 4
  1.5 Overview of Thesis 4

II. Basic Concepts and Architectures 6
  2.1 Correlation 6
  2.2 PSRI Feature Space 7
  2.3 Horev Algorithm 10
  2.4 Optical Image Processing 12
  2.5 Miazza Architecture 14
  2.6 Summary 16

III. Extended Analysis of Optical Implementation and Digital Simulation 17
  3.1 Introduction 17
  3.2 Defining Images 17

Page

3.3 Fourier Transforms 20

3.3.1 Background 20

3.3.2 Analysis 22

3.4 Polar-Logarithmic Coordinate Transformation Using a CGH 27

3.4.1 Introduction 27

3.4.2 Optical Implementation 27

3.4.3 Alignment and Focusing 28

3.4.4 Space-Bandwidth Product 34

3.4.5 Linearity 34

3.4.6 Data Storage 41

3.5 Correlating The Feature Space 41

3.5.1 Vander Lugt Filtering 41

3.5.2 Thermoplastic Holograms 45

3.5.3 *Experimental* Setup for Correlations 49

3.6 Digital Simulations 51

3.7 Summary 54

IV. Experimental Results 58

4.1 Introduction 58

4.2 Optical FLRT Feature Space Results 58

4.3 Correlation Results 64

4.3.1 Target Rotations 73

4.3.2 Target Scale 82

4.3.3 Target Detection 83

4.4 Summary 93

Page

V. Conclusions and Recommendations 94

5.1 Introduction 94

5.2 Conclusions 94

5.3 Recommendations 96

Appendix A. DC-Block and Image Template Preparation 99

A.1 Introduction 99

A.2 DC-Blocks 99

A.3 Preparation 100

A.3.1 The Dekacon 100

A.3.2 The Procedure 101

A.3.3 Templates 102

Appendix B. The Computer Generated Hologram 104

B.1 Generation 104

B.2 The Metalib Program 105

Appendix C. Sight-Mod Spatial Light Modulator 109

Appendix D. CCD Camera Image Format 115

D.1 CCD Images 115

Appendix E. Code to Generate the ϕ Function 131

Bibliography 133

Vita 135

Vita 136

List of Figures

Figure Page

1. Casasent and Psaltis' PSRI Algorithm 7
2. (a) Input Square. (b) 167% Scale Change and 45° Rotation. 8
3. Theoretical $\ln r - \theta$ CT of a square. 9
4. Horev Algorithm 11
5. Miazza's Optical Pattern Recognition System. 15
6. Setup for imaging binary templates onto CCD camera. 19
7. Photograph of small square stored on diskette. 20
8. Schematic of optical Fourier transform setup. 21
9. Comparison of FTs of a square. 24
10. Optical Fourier transforming setups. 26
11. Mayo's $f_{\ln r} - f_\theta$ feature space setup. 29
12. Optical setup for $f_{\ln r} - f_\theta$ CT. 30
13. Photograph of the $\ln r - \theta$ CT of a small square. 31
14. Out-of-Alignment CTs 32
15. Alignment of CTs 33
16. Out-of-focus CTs 35
17. CTs of circular apertures 36
18. CTs of circular apertures 37
19. CT of square rotated 45 degrees. 38
20. Optical setup to test linearity of $f_{\ln r} - f_\theta$ CGH. 39
21. Table and Graph of aperture size versus distance from zero reference. 40
22. Optical Schematic for generating Vander Lugt filter. 42
23. Optical schematic to generate optical correlation. 44
24. Cross section of thermoplastic film. 46

Figure Page

25. Four step development process for generating a thermoplastic hologram. 47

26. Coronatron charging a thermoplastic plate. 48

27. Optical setup for correlating the FLRT feature space. 50

28. Optical autocorrelation of the CT of a square. 52

29. CGH artifact. 53

30. Digitally simulated optical $f_{\ln r} - f_\theta$ transform of optical FT of small rotated square. 55

31. Digital correlation of FLRT feature space. 56

32. Development of FLRT CT of a square. 60

33. Development of FLRT CT of a square scaled by a half. 61

34. Development of the FLRT CT of a square scaled by a half and rotated by 45° 62

35. Development of FLRT CT of a tank. 65

36. Development of FLRT CT of a tank scaled by 3/4. 66

37. FLRT CT of a truck. 67

38. Development of FLRT CT of an F-15. 68

39. Development of FLRT CT of a truck and tank rotated 90° with respect to each other. 69

40. Development of FLRT CT of two tanks and two trucks. 70

41. Development of FLRT CT of a tank, truck , and F-15. 71

42. Development of FLRT CT of three F-15's. 72

43. Optical autocorrelation of the CT of a square. 74

44. Digital autocorrelation of the CT of a square. 75

45. Optical correlation of the CTs of a scaled and rotated square and a square. 76

46. Digital correlation of the CTs of a scaled and rotated square with a square. 77

47. Optical correlation of the CTs of a truck and of a tank and rotated truck. 78

Figure Page

48. Digital correlation of the CTs of a truck and of a tank and rotated truck. 79

49. Graphic illustration of 4π versus 2π optical correlation. 81

50. Optical correlation of the CTs of a truck and multiple vehicles. . . . 84

51. Optical correlation of the CTs of an F-15 and three F-15s. 85

52. Optical correlation of the CTs of a square and a square, triangle, and circle. 86

53. Optical correlation of the CTs of a truck and an F-15. 88

54. Optical correlation of the CTs of an F-15 and a tank. 89

55. Optical correlation of the CTs of a square and a tank. 90

56. Optical autocorrelation of the CT of an F-15. 91

57. Optical correlation of the CTs of an F-15 and a scaled F-15. 92

58. Dekacon Camera Setup 103

59. Laser printer plot of a CGH. 106

AFIT/GE/ENG/88D-4

Abstract

This thesis investigates the feasibility of optically implementing a real-time, pattern recognition system using correlation techniques in a position, scale, and rotation invariant (PSRI) feature space. Input target templates were optically Fourier transformed using an improved, high-resolution, high-pass filter positioned in the focal plane of the transforming lens. A logarithmic-polar coordinate transform of the magnitude-squared of the Fourier transform ($|FT|^2$) was performed with an improved method of focusing the magnified $|FT|^2$ onto a computer generated hologram (CGH), which was shown to scale linearly along the horizontal axis and logarithmically along the vertical axis. Optical, matched-filter correlations on the magnitude-squared Fourier transform logarithmic-polar (FLRT) feature space were performed using thermoplastic, phase-relief holography and Vander Lugt filtering.

Correlation results prove that scale and rotation changes of the input can be predicted accurately based on linear shifts of the correlation peak. Also, the FLRT feature space is shown to provide excellent discrimination for multiple-input scenes. The need for a cyclic correlation is verified, and digital simulations prove the validity of the optical results.

POSITION, SCALE, AND ROTATION INVARIANT
OPTICAL PATTERN RECOGNITION
FOR
TARGET EXTRACTION AND IDENTIFICATION

I. Introduction

For many years man has sought an algorithm that would enable him to develop a machine that can detect, locate, and identify targets. Obviously, there are numerous applications for such a machine. For example, machine vision could allow a robotic system to maneuver in a dynamic environment while performing aircraft maintenance on a flight line. Machine vision could also be used in designing autonomous weapons that seek out specific targets. The military has a need to develop just such autonomous systems to detect and classify potential targets for use in missile guidance and strategic surveillance [9:1]. However, autonomous pattern recognition systems must be able to accurately detect and classify targets in real-time and in the presence of scene irregularities like background noise, illumination variations, and variations in target size, position, and in-plane rotation. While much research has gone on in both digital and optical techniques for pattern recognition, a truly autonomous, real-time system which accurately classifies targets embedded in their natural surroundings has not been developed.

1.1 Background

Pattern recognition, under all but the most restrictive conditions, is a very unstructured field. Potential targets in a scene may appear in various sizes, rotations, or even be partially occluded by other objects within the scene. With the overwhelming number of possible variations, using digital techniques alone, even with the most sophisticated algorithms, becomes unmanageable when only a few variables are allowed to change. Since optical systems process entire scenes in parallel and do so in the time it takes for the light to travel through the system, optical pattern recognition has become increasingly more popular and promising [9].

Correlation between an input image (scene) and a template is the basic approach used in most optical (as well as digital) pattern recognition algorithms [9:2]. However, correlation, when implemented using Fourier transforms, is not a scale invariant transformation. Casasent and Psaltis have shown that coherent, optical correlations which are both scale and shift invariant can be accomplished using a combined Fourier-Mellin transform [3, 4]. With this Fourier-Mellin technique, in conjunction with Vander Lugt filtering [6:171–184], Casasent and Psaltis have developed a position, scale, and rotation invariant (PSRI) transformation for objects in pristine backgrounds [3:78].

In 1980, Horev, during his thesis work at the Air Force Institute of Technology (AFIT), developed an algorithm which could be used for locating objects in two-dimensional intensity images without prior knowledge of position, scale, or rotation of the object [7]. Horev's algorithm used an approach similar to Casasent and Psaltis to demonstrate recognition with translation, scale, and rotation invariance in high clutter scenes. Kobel and Martin, in 1986, digitally implemented Horev's algorithm on the VAX 11/780 system at AFIT. Using modified phase correlation and a Blackman smoothing function, they also improved Horev's algorithm to enhance its ability to determine correlation peaks needed for target classification and extraction [9]. The algorithm is now known as the Kobel-Martin-Horev algorithm.

In 1987, Mayo attempted an optical implementation of portions of the Kobel-Martin-Horev algorithm using a computer generated interferogram (also known as a computer generated hologram (CGH)) and a magneto-optic spatial light modulator (SLM) [12]. Detailed descriptions of the CGH and the SLM are given in Appendices B and C, respectively. He successfully implemented a PSRI feature space but did not complete optical implementation of the matched-filter correlations. Also in 1987, Miazza improved and extended the Kobel-Martin-Horev algorithm and developed a model for the optical implementation of the improved algorithm. The essence of Miazza's work involves phase extraction and reinsertion using optical methods [14].

Troxel used Kobel and Martin's improved version of Horev's algorithm to implement target recognition using range data and neural networks [18]. He used segmented images of tanks, trucks, and jeeps in a digital implementation of Kobel and Martin's algorithm to prepare autocorrelation and cross-correlation peaks of the PSRI space. The segmented images, both hand segmented and computer segmented, were binary silhouette templates created from the actual range images [18:Sec II,1]. Using a multilayer perceptron implementation of a neural network, Troxel attempted to classify random input scenes of trucks, tanks, and jeeps by training the neural network with samples of autocorrelations and crosscorrelation of the objects [18:Sec VI,6]. The neural network was also a digital implementation and not an actual hardware network. Although Troxel's work was successful in proving the validity of the algorithm, the classification of targets was still done digitally and its implementation was far from real-time.

1.2 Problem Definition

The thesis problem is to investigate the hybrid pattern recognition work of Kobel and Martin and of Mayo and Miazza in an effort to determine the feasibility of a real-time, optical pattern recognition system. The system would be invariant to the position, scale, and rotation changes of objects in realistic input scenes and could be used to locate and classify targets in real-time.

1.3 Scope of Thesis

This thesis focuses on investigating the optical implementation of correlation techniques used in extracting and identifying targets from raw input scenes. This was accomplished by first readdressing and improving upon the work of Mayo through improved optical techniques, and then by demonstrating and evaluating portions of the Miazza design. In particular, Vander Lugt filter correlations of targets in the magnitude-squared Fourier transform $\ln r - \theta$ feature space were constructed and evaluated.

1.4 Approach

To prepare for the testing of the individual steps in the Miazza design, a set of target scenes and templates were generated. The scenes and templates ranged in complexity from simple geometric figures to actual photographs of tanks and trucks. In template preparation, these images were transferred to high-resolution. glass plates for optical implementation. They were also stored in digitized files on the VAX 11/780 computer in the Air Force Institute of Technology's engineering building. The software package "Executive" developed by Kobel and Martin [9:Vol II] to digitally implement the Kobel-Martin-Horev algorithm verified the accuracy of the optical data obtained during this research.

Correlation of the feature space is demonstrated in three phases. First, the target scenes and templates are Fourier transformed and then $\ln r - \theta$ coordinate transformed. Next, a Vander Lugt filter of the templates is made with a thermoplastic holographic camera. Finally, the Vander Lugt filter is used to produce correlations of the transformed target scenes with the transformed target templates. The correlation peaks are analyzed for accuracy of indicating target scale and rotation angle.

1.5 Overview of Thesis

This thesis is structured to present the material in the natural order that the investigation progressed. The present chapter introduced the basic problem to be addressed and then quickly summarized some of the background work which supports this thesis effort. In addition, this chapter explained the scope of the thesis, as well as, the approach taken to solve the problem.

Chapter II discusses the theoretical background of many of the basic optical concepts investigated during this thesis effort and previously mentioned in the *Background* section of this chapter. These concepts include: correlations, Fourier transforms, coordinate transforms, CGHs, and matched-filtering. Additionally, the Horev algorithm and the Miazza architecture are detailed and explained.

Chapter III details how the optical concepts introduced in Chapter II are used during this thesis effort. Experimental setups for implementing these optical concepts are shown and explained. Additionally, the digital simulations performed to verify the optical results are discussed.

Chapter IV explains and analyzes the results of both the optical experiments and the digital simulations. Photographs of both $f_{\ln r} - f_{\theta}$ coordinate transformations and correlation peaks are presented.

Finally, Chapter V presents pertinent conclusions based on the available data. In addition, recommendations for furthering this research, including suggested experiments, are made.

II. Basic Concepts and Architectures

2.1 Correlation

One concept that has been given much attention in developing a pattern recognition algorithm is correlation [8]. Correlation is a mathematical comparison of two signals (or in this case images) that measures the degree of similarity between the shape of one image to another. The advantage of correlating an image $f(x,y)$ with a target template $t(x,y)$ is that the result will be a distinguishable autocorrelation peak when the image and the target are the same, and an equally distinguishable crosscorrelation peak when the image and the template are dissimilar. Mathematically, correlation can be described by [6:78]

$$\begin{aligned} c(x,y) &= f(\alpha,\beta) \star t(\alpha,\beta) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)t^{*}(\alpha - x, \beta - y)d\alpha d\beta \end{aligned} \tag{1}$$

where α and β are dummy variables of integration, the asterick $(*)$ means complex conjugate, the star $(\star)$ is the mathematical symbol for correlation, and $c(x,y)$ is the correlation result. Alternatively, correlation can be expressed in terms of the Fourier transform as follows[6:10]

$$c(x,y) = \mathcal{F}^{-1}\{FT^{*}\} \tag{2}$$

where $\mathcal{F}^{-1}$ is the symbol for an inverse Fourier transform operation and F and T are the Fourier transforms of $f(\alpha,\beta)$ and $t(\alpha,\beta)$, respectively. Another advantage of correlating signals is that an autocorrelation results in a peak positioned relative to the location of the target. Furthermore, correlators will detect more than one target in a scene [8].

The major disadvantage of correlation is that the performance is severely degraded by both scale and rotation differences between the template and the target. Casasent and Psaltis reported that the signal-to-noise ratio (SNR) of a correlation

$$\begin{tikzcd}
f(x,y) \arrow[r, "FT"] & {|F(f_x, f_y)|} \arrow[r, "POLAR\ CONVERSION"] & {|F(f_r, f_\theta)|} \arrow[d, "\ln r"] \\
\text{PSRI FEATURE SPACE} & & {|F(f_{\ln r}, f_\theta)|} \arrow[ll, "FT"']
\end{tikzcd}$$

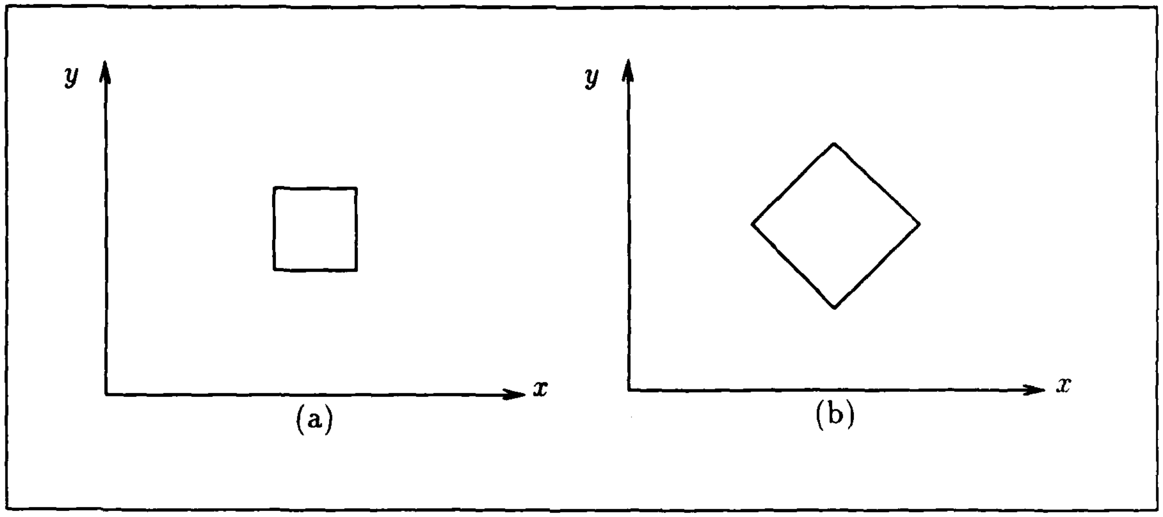
Figure 1. Casasent and Psaltis' PSRI Algorithm

peak would drop 27 decibels (dB) when the scale of the target and template differed by 2 percent or when the target was rotated 3.5 degrees with respect to the template [3:77].

2.2 PSRI Feature Space

In 1978, Casasent and Psaltis, suggested that the scene space, which is the domain of the original image, be transformed into a feature space that is invariant to changes in a target's scale and rotation [2:339–340]. In such a feature space, a correlation could be performed without loss of SNR of the autocorrelation peak. The key questions here are whether this feature space exists, and how much information about the target will be lost when a target is transformed into this feature space?

Casasent and Psaltis proposed a feature space that was position, scale, and rotation (in-plane) invariant (PSRI). To derive a PSRI feature space, their algorithm uses four separate transformations. The transformation sequence is outlined in Figure 1 [2:339–345]. The first step, taking the Fourier transform (FT) of the scene



(a)

(b)

Figure 2. (a) Input Square. (b) 167% Scale Change and 45° Rotation.

space, utilizes the shifting property of the Fourier transform [6:9]:

$$f(x \pm x_o) \xrightarrow{FT} F(f_x)(\exp(\pm j2\pi x_o f_x)) \tag{3}$$

This property states that the Fourier transform of a shifted signal $f(x)$ is equal to the Fourier transform of $f(x)$ multiplied by a linear phase term [6:277]. Therefore, taking the magnitude of the Fourier transform ($\mid F(f_x, f_y) \mid$) eliminates the phase term and the positional dependence. Optically, this means that the Fourier transform of an image will always be centered on the optic axis of the Fourier plane. Then, if only intensity (which is the magnitude squared) is considered, the Fourier transform of a shifted image will look exactly like that of the unshifted image.

The second step in the algorithm is to perform a polar coordinate transformation on $\mid F(f_x, f_y) \mid$ yielding $\mid F(f_r, f_\theta) \mid$ where f_r represents the distance in the radial direction and f_θ represents the angle in the rectangular coordinate system. This transformation converts rotational shifts about the origin of the rectangular frequency domain into linear shifts along the f_θ axis of the polar frequency domain (see Figures 2 and 3).

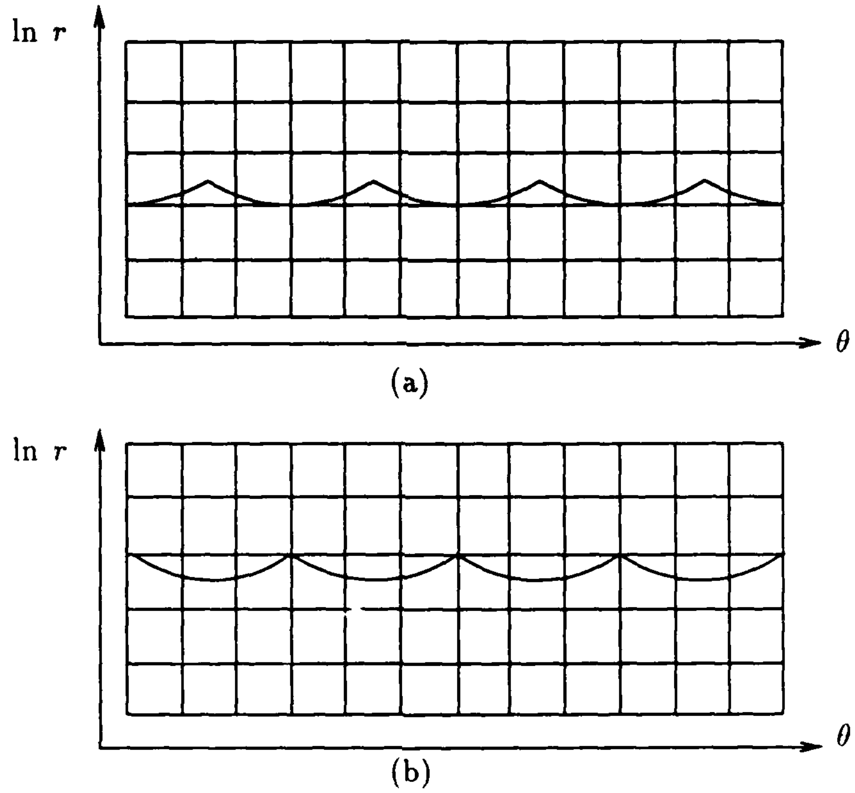


Figure 3. Theoretical $\ln r - \theta$ Coordinate Transform of Squares in Figure 2 (a) and (b). Note: The transform in (b) is shifted up and to the right.

The third step is to logarithmically (ln) scale the radial axis of the polar transformation. This transformation converts scale changes in the rectangular FT space into linear shifts of $| F(f_r, f_\theta) |$ along the radial axis. Consider the FT scaling property [6:9]

$$f(x/a, y/a) \xrightarrow{FT} a^2 F(af_x, af_y) \xrightarrow{r-\theta} a^2 F(af_r, f_\theta) \tag{4}$$

where a is a scaling constant. Taking the natural logarithm of the radial component yields [9:9]

$$\ln(af_r) = \ln(f_r) + \ln(a) \tag{5}$$

As demonstrated in Figures 2 and 3, the logarithm converts the scale difference in the rectangular domain into a linear shift on the $f_{\ln r}$ axis.

In the final step of the Casasent algorithm, the FT of $\mid F(f_{\ln r}, f_{\theta}) \mid$ converts the linear shifts along the $f_{\ln r}$ and f_{θ} axes into phase terms that can be eliminated with an intensity detector [2:342]. Performing template correlation in Casasent and Psaltis' PSRI feature space will identify the presence of a target. However, it will not be able to provide the parameters necessary for target acquisition: position in the field of view, scale, and rotation.

2.3 Horev Algorithm

In 1980, Horev developed a pattern recognition algorithm [7] that provided the foundation for the future development of a real-time pattern recognition machine at the Air Force Institute of Technology (AFIT). A flow chart of the algorithm is shown in Figure 4. The algorithm utilizes the previous architecture explored by Casasent and Psaltis. The beauty of his design is two-fold. First, he performs his correlation at step 3 in the Casasent and Psaltis algorithm (see Figure 1). Performing the correlation in the $\mid F(f_{\ln r}, f_{\theta}) \mid$ feature space preserves the target's scale and rotation information. Now the target's scale and rotation can be specified by determining the position of the target-template autocorrelation peak relative to a predetermined reference point. The scale and rotation information is then used to generate a rectified template that has the same scale and rotation as the target. Horev accomplished this by shifting the template feature space by an appropriate amount along the $f_{\ln r}$ and f_{θ} axes.

Second, Horev introduces a second stage that extracts the phase information from the FT of the target. The phase is added to the rectified template and inverse Fourier transformed to yield a rectified target image. Since the phase contains the target's positional information, and the rectified template contains only the frequency energy of the target, the target's scene clutter (noise) will be reduced and the

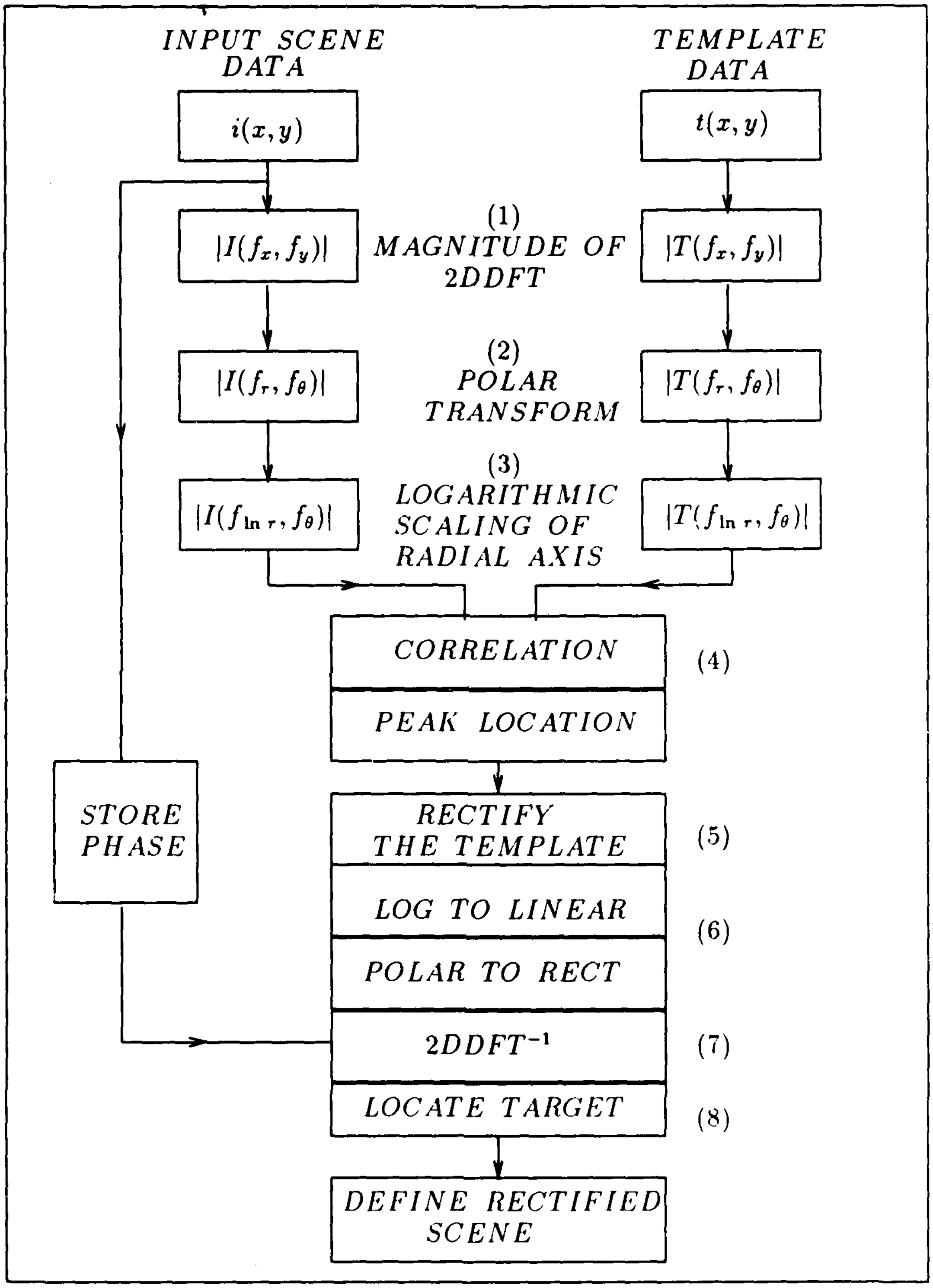


Figure 4. Horev Algorithm. Note: The addition of the phase extraction and reinsertion stages allows for target segmentation.

target enhanced [7, 9]. Horev uses the rectified target image and a rectified template to perform a correlation in the scene space to determine the location of his target. Since the template has been adjusted to the target's scale and rotation (in-plane), the algorithm has sought to maximize the SNR of the autocorrelation peak [7:87–88].

Horev digitally implemented his algorithm and found that it had definite promise in high clutter rejection [7:87]. However, the computer implementation took over four hours to process one scene [7:89]. Kobel and Martin conducted a follow-on study with Horev's algorithm [9:9]. They developed another digital implementation that reduced the processing time down to about half an hour per scene [8]. They found that the algorithm recognized multiple targets in a broad range of non-random, noise corrupted scenes [9:78–79]. However, the problem with the intense computations associated with the Fourier-transformed, coordinate-transformed correlations remained. To overcome this, the feasibility of optical implementation needed to be investigated.

2.4 Optical Image Processing

Fourier transforms, coordinate transformations, and correlation schemes have been heavily scrutinized as potential pattern recognition tools since they can be implemented optically and significantly reduce the processing time [9]. Using these tools, the goal is to design an optical pattern recognition system that can provide parallel processing at the speed of light.

The Fourier transformation is a fundamental building block in optical image processing. Placing an image in the input focal plane of a lens will yield the Fourier transform of the image in the output focal plane [6:83–90]. Thus, the first step in the Kobel-Martin-Horev algorithm is easily accomplished optically by sensing the intensity of the Fourier transform of the input data. The Fourier transform property is also used to design computer generated holograms (CGH) to perform coordinate transformations. The coordinate transformation is developed from the following [1, 16]

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp(j\phi(x,y))\exp[-j(2\pi/\lambda f_L)(xu+yv)]dxdy \qquad (6)$$

where

$F(u,v)$ = coordinate transform of $f(x,y)$

λ = wavelength of light

f_L = focal length of the Fourier transforming lens

This equation states there is a phase function $\exp[j\phi(x,y)]$ that can be multiplied by an image $f(x,y)$ and Fourier transformed to yield a specified coordinate transformation.

For the $f_{\ln r} - f_\theta$ coordinate transformation, Eq 6 can be solved for $\phi(x,y)$ using [1:939]

$$u(x,y) = \ln(x^2+y^2)^{1/2} \qquad (7)$$

$$v(x,y) = -\arctan(y/x) \qquad (8)$$

and the approximate saddle point integration method. This method yields [16:9]

$$\phi(x,y) = \frac{2\pi}{\lambda f_L}[x\,\ln(x^2+y^2)^{1/2} - y\arctan(y/x) - x] \qquad (9)$$

A CGH is a binary interferogram of the interference pattern of $\phi(x,y)$ and a plane wave at an angle θ. The lines of the CGH must be plotted to satisfy [1:939]

$$2\pi\alpha x - \phi(x,y) = 2\pi n \qquad (10)$$

where n is the integer that indicates different lines, and $\alpha = \sin\theta/\lambda$. Once the PSRI feature space has been created, the optical correlation must be accomplished.

Optical correlation is accomplished using a Vander Lugt filter [6:171-183]. A Vander Lugt filter has a transmittance proportional to the Fourier transform of the template that is amplitude and phase modulated with a high frequency carrier.

The filter is produced by exposing high resolution film directly with the Fourier transform of a template, while simultaneously exposing the film at an angle with a coherent, plane wave. When the developed filter is illuminated directly with the Fourier transform of a target image, the output, when inverse Fourier transformed, will be the correlation of the target with the template traveling away from the filter plane at the same angle the filter was originally exposed with the plane wave.

2.5 Miazza Architecture

In 1988, Miazza introduced a design for a real-time coherent optical pattern recognition system that implements the Horev-Kobel-Martin algorithm [13]. A schematic of his design is shown in Figure 5. The key to his design is the use of spatial light modulators (SLM), optical devices that control the output intensity of an incident light pattern, to perform computationally intensive image processing at real-time rates.

Miazza uses SLMs to perform four different processes in his system: incoherent-to-coherent light (ITC) conversion, phase extraction, and phase insertion. In Figure 5, SLM 1 functions as an ITC converter for the input image. This allows the system flexibility in the type of input images it can process. SLM 2 and SLM 4 both remove the phase of the image and allow the magnitude to pass through the system for processing. Phase removal is very important since the $\ln r - \theta$ CGH and the Vander-Lugt filter F3 are both designed to process only the magnitude of an image. SLM 3 and SLM 5 are the highlight of the system because they perform a very difficult task in optics—the phase extraction and reinsertion step of the Horev-Kobel-Martin algorithm (see Figure 4). Miazza proposes that a SLM be built (SLM 3) with either a transmittance or a reflectance inversely proportional to the intensity of the input image. Therefore, the magnitude of the output image will be unity, and the output will retain the phase information of the input. A separate solution, using nonlinear photorefractive crystals, has also been proposed [19]. SLM 5 takes the

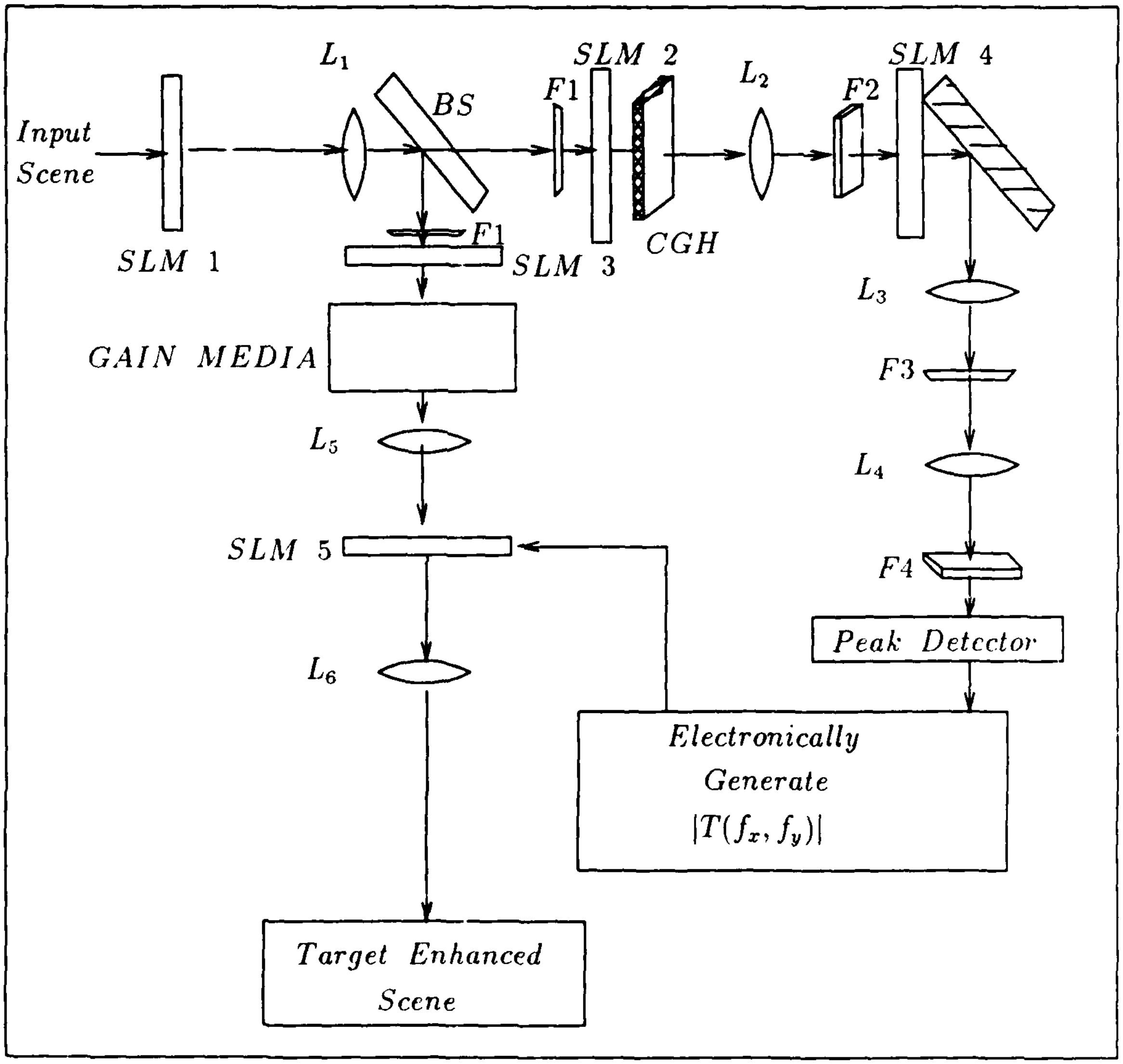


Figure 5. Miazza's Optical Pattern Recognition System.

phase information from SLM 3 and places the magnitude information of the rectified template on the same image.

Miazza's design is dependent on spatial light modulators that are still being designed and tested. Consequently, only a portion of his system has been validated in a laboratory. Casasent, Saito, and Mayo have fabricated and tested a $\ln r - \theta$ CGH interferogram [1, 10, 12, 16]. However, in each case only the scene space was transformed instead of testing coordinate transformation of the position invariant magnitude-squared Fourier transform space.

2.6 Summary

Optical techniques may be the key in the development of a real-time pattern recognition machine. Pattern recognition algorithms that utilize Fourier Transforms are ideal for optical processing since they can be implemented with one lens. Miazza has designed a real-time optical pattern recognizer based on an algorithm developed by Horev. The design uses spatial light modulators, lenses, and a computer generated hologram. These techniques need to be tested in a laboratory environment.

III. Extended Analysis of Optical Implementation and Digital Simulation

3.1 Introduction

This chapter details the experiments performed and analyses made while implementing portions of the Kobel-Martin-Horev (KMH) algorithm optically. Additionally, a section of the chapter is dedicated to the digital simulations that were performed in parallel with the optical experiments to verify results obtained optically.

The next section of this chapter discusses the initial imaging experiments required to prepare a set of binary templates and reproduceable stored images. This is followed by a section on Fourier transforming, including setups used and problems encountered. Next is a section on the computer generated hologram (CGH). detailing how it was used to produce the $f_{\ln r} - f_{\theta}$ coordinate transformation and problems associated with linearity and space-bandwidth product (SBWP). A section on optical correlations follows. In this section, details of the Vander Lugt filter and thermoplastic holography are given. The next to last section details the digital simulation, and the last section summarizes the important highlights of this chapter.

3.2 Defining Images

After becoming more familiar with the equipment available for this research, the researchers' first order of business was to decide upon a set of images to use as binary templates throughout the research. The set agreed upon consisted of the following:

1. Simple geometric images to include a square, an edged square, and a triangle.
2. Geometric combinations to include a square, triangle, and circle; and an edged square, triangle, and circle.

3. Letters to include an "A" and an "ASL."
4. Cutouts of tanks, trucks, and F-15s.

It should be noted that each of the images listed above was produced in at least two different scales and could be rotated and translated at will within the optical systems. This scale, rotation, and translation variation allowed this research to address three of the six problem areas identified by Kobel and Martin [9:8].

Simple geometric images were chosen as research images because the symmetry of these objects allowed us to predict, ahead of time, the shape of their Fourier transforms and coordinate transforms. Using them would then allow us to have confidence in the optical implementation if observed results were commensurate with predicted results. Letters were chosen because much pattern recognition work has centered around recognizing the written word. Finally, tanks, trucks, and F-15s were used because segmented images were readily available, and the military has a need to be able to identify them as potential targets.

Once the desired images had been identified, the templates had to be produced. An initial attempt was made at photographing (using a 35mm camera) a white cutout on black background using instant slide film. However, results were less than desirable (see Appendix A). To get high quality, binary templates, high resolution glass photographic plates were used. Details of how the binary templates were produced on the glass plates can be found in Appendix A.

With a set of high quality, binary templates now in hand, research could proceede. Knowing that digital manipulations would be performed on the templates, a means of transferring them to the VAX I780A had to be devised. The first step was to get the images stored on floppy diskettes. The optical setup shown in Figure 6 is the setup used to store the images. The two lens imaging system was set up for an image reduction of 2.5 times. This allowed the entire image of even the largest template to be stored successfully. Filter $F1$ was actually as many neutral

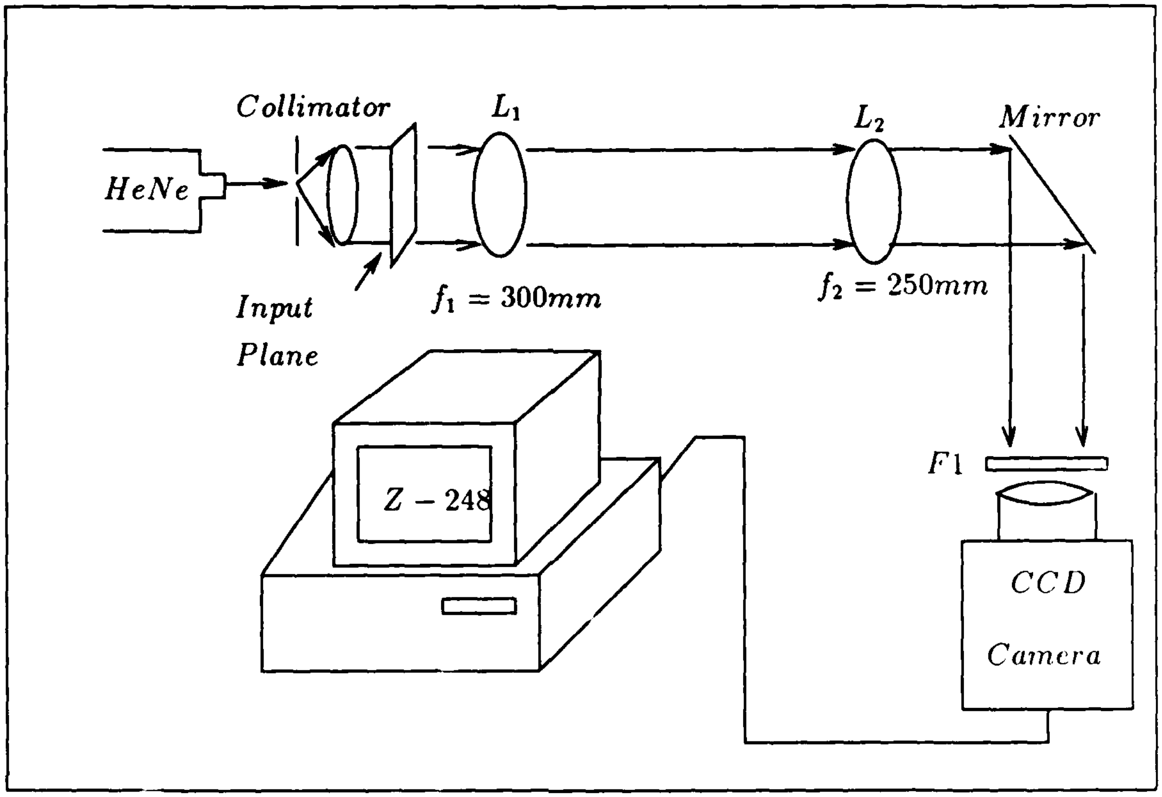


Figure 6. Setup for imaging binary templates onto CCD camera. Note: Used to store images on disk at 2.5x reduction.

density filters as necessary to reduce background illumination to provide as binary an image as possible. Templates were placed in a filter holder and illuminated with the coherent laser light source. The CCD camera, driven by the AT&T TARGA framegrabber, then grabbed the images. Once grabbed into the TARGA's memory, the images could be transferred to floppy disk. The images were then transferred to the VAX using the file transfer protocol (ftp) routine. This procedure is explained in detail in Appendix D. Figure 7 shows a photograph of the same small square shown above after it was saved on disk and transferred to the VAX.

The next step of the KMH algorithm called for taking the magnitude of the Fourier transform ($|FT|$) of the images. Optical implementation of this step is

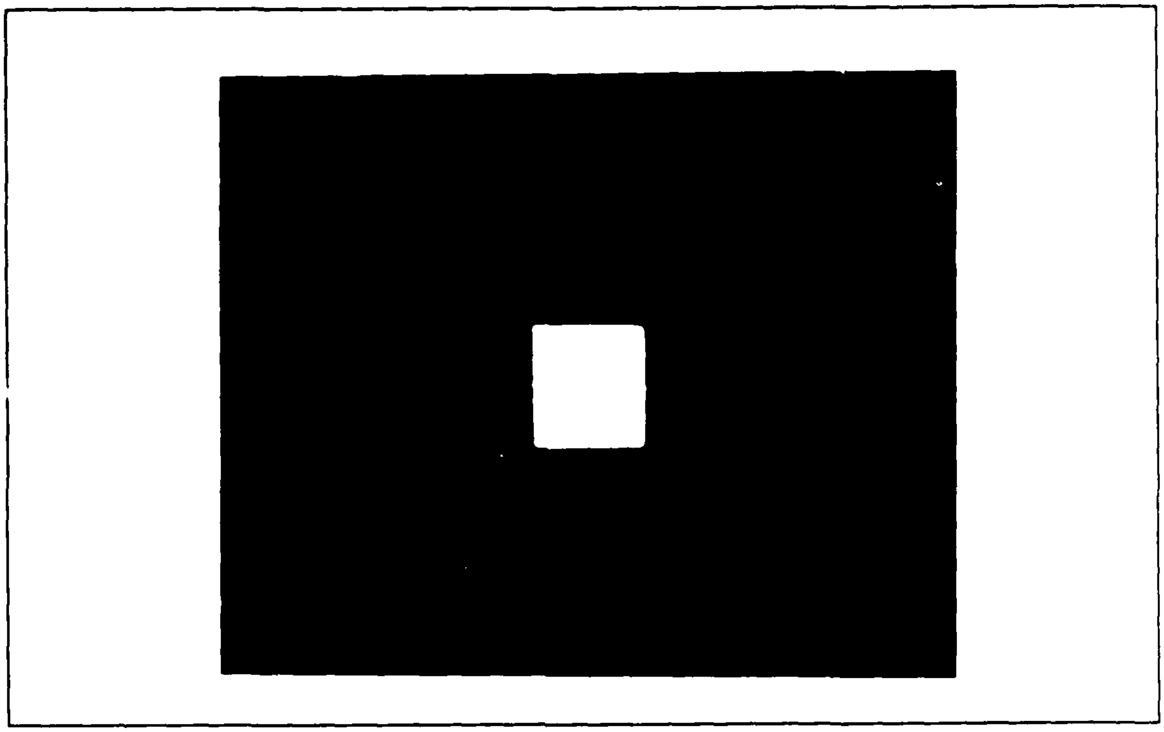


Figure 7. Photograph of small square stored on diskette. Note: Taken from display of a microVAX workstation.

detailed in the following section.

3.3 Fourier Transforms

3.3.1 Background The very essense of any optical lens system is the inherent ability of a positive (converging) lens to produce the Fourier transform of a coherent input. Therefore, it is no surprise that the initial image transformation in this thesis work, as it was in the KMH algorithm, is the Fourier transform (FT) of the input image scene. While this step alone is the limiting time factor that hinders the KMH algorithm from being implemented in near real-time, it can be realized instantaneously in an optical system. This is a very distinct advantage of implementing the KMH algorithm optically.

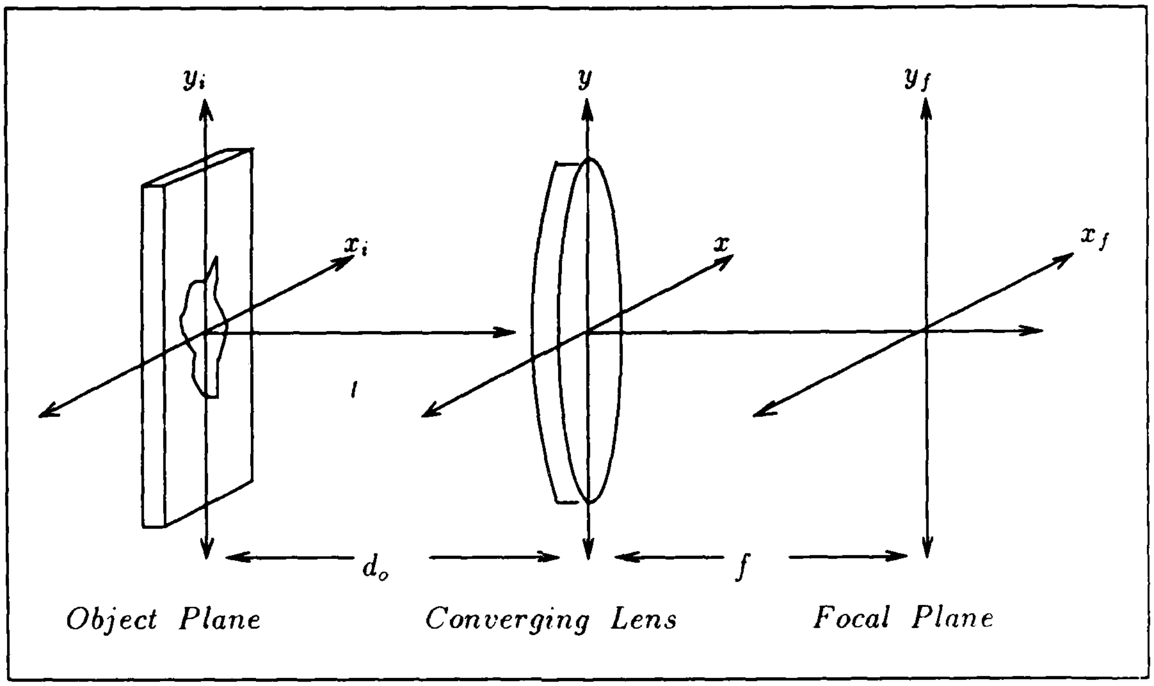


Figure 8. Schematic of Optical Fourier Transform Setup [6:87] .

Kobel and Martin used a two-dimensional, fast Fourier transform algorithm ($2DFFT$) for computing their initial FT . This method took approximately a minute to calculate the FT of a 256 by 256 pixel image array using AFIT's I780a VAX. With the typical optical setup shown in Figure 8, FTs are generated at the speed of light. When image $t(x_i, y_i)$ is placed at a distance d_o in front of a converging lens, the output image $U_f(x_f, y_f)$ located at the lens' focal plane (one focal length behind the lens) has a direct relationship to the FT of the input image. This relationship is described by the following equation [6:86]:

$$U_f(x_f, y_f) = \frac{A \exp[j \frac{k}{2f}(1 - \frac{d_o}{f})(x_f^2 + y_f^2)]}{j2\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} t(x_i, y_i) \exp[-j \frac{2\pi}{\lambda f}(x_i x_f + y_i y_f)] dx_o dy_o \qquad (11)$$

where

f = focal length of the converging lens

λ = wavelength of coherent light
d_o= distance of input in front of the lens
k = wave number $(\frac{2\pi}{\lambda})$
A = amplitude constant for light source

This equation states that the amplitude and phase at the focal plane (x_f, y_f) are related to the amplitude and phase of the frequency spectrum of $t(x_i, y_i)$ at the frequency $(x_f/\lambda f, y_f/\lambda f)$ [6:86]. The output $U_f(x_f, y_f)$ is actually the amplitude-scaled FT of the image multiplied by a phase function Φ . Therefore,

$$U_f(x_f, y_f) = T(\frac{x_f}{\lambda f}, \frac{y_f}{\lambda f})\,\Phi \tag{12}$$

where

$T(\frac{x_f}{\lambda f}, \frac{y_f}{\lambda f})$ = FT of the input image
$\Phi = \frac{A \exp[j \frac{k}{2f}(1 - \frac{d_o}{f})(x_f^2 + y_f^2)]}{j2\lambda f}$

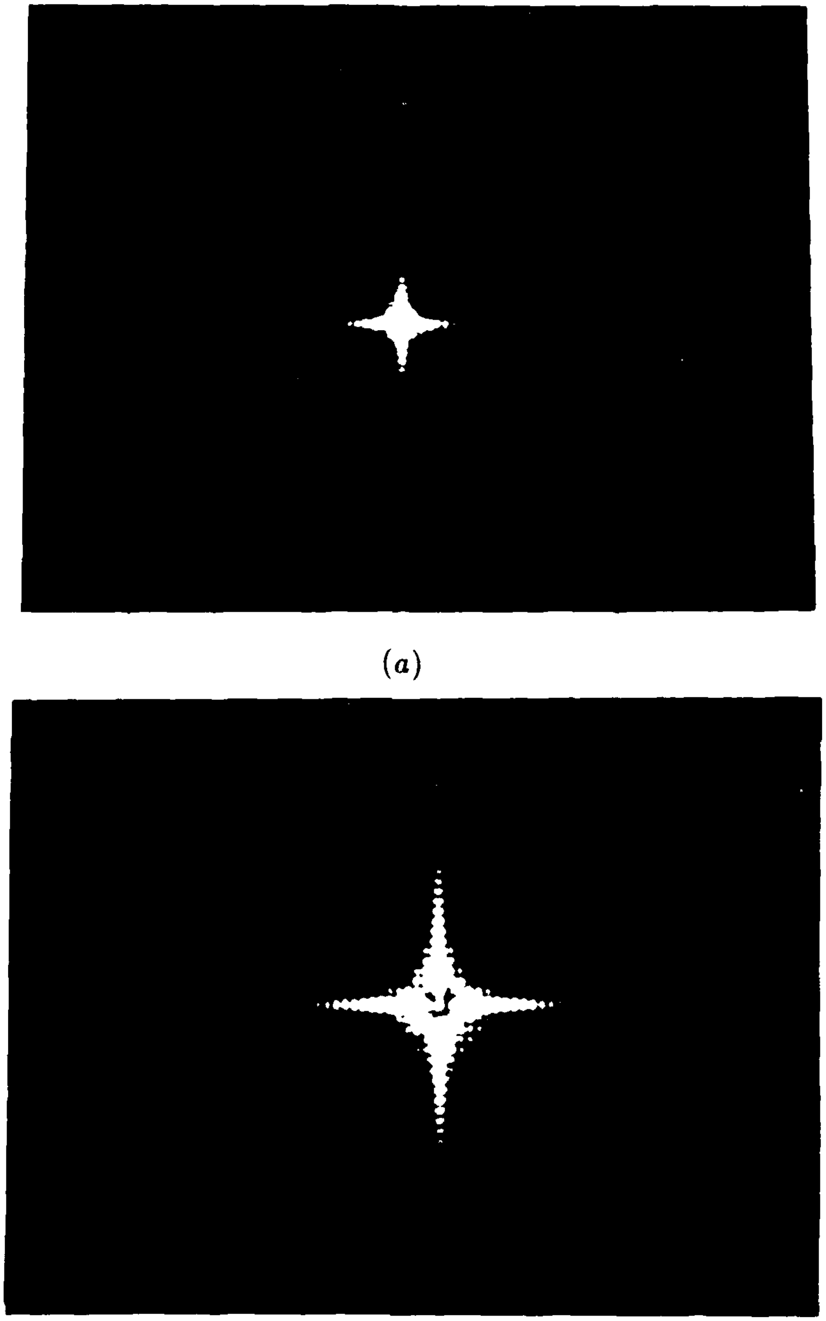
Notice that when $d_o = f$, Φ is equal to a constant $A/j2\lambda f$ and $U_f(x_f, y_f)$ is an exact FT of $t(x_o, y_o)$ scaled by the constant $A/j2\lambda f$.

3.3.2 Analysis In this research, a helium neon (HeNe) laser was used as the coherent light source, and a 300 mm focal length, f , converging lens was used as the Fourier transforming lens. Since the computer generated hologram (CGH) used in the second transformation of the AFIT algorithm can transform only real inputs, the phase information must be removed from the Fourier transformed image. This can be accomplished by one of two methods: first, using a charged coupled device (CCD) camera with a framegrabber board; and second, recording the image directly on high resolution, black and white 35 mm film.

Of the two techniques, using the CCD camera with a framegrabber is the only technique usable in a real time optical system. Actually, to build a continuous system that detects the input image's intensity and retransmits the intensity distribution without its phase would also require a spatial light modulator (SLM). Miazza proposed that a grey level sensitive optically addressed SLM be used for real time phase removal [13]. Since this type of SLM was not available for study, we used a 512 by 512 element Sony model XC-38 CCD camera and a AT&T TARGA framegrabber board installed in a Zenith 248 (IBM AT compatable) personal computer. The CCD camera detected intensity of the image signal. The detected image intensity was then digitized and stored in memory by the TARGA board. To retransmit the image intensity, we used a Semetex 128 by 128 pixel Sight Mod. This is a magneto-optical binary SLM. Since this device was binary, that is, no grey scale, it limited the use of this technique to showing that image intensities can be detected and retransmitted at near real time. For further discussion of the Semetex Sight Mod see Appendix C.

Mayo discovered that the CCD camera was very sensitive to high intensity peaks like the dc (or center) of a *FT* [12]. When the CCD array was exposed to a high intensity peak, the camera exhibited a "blooming" phenomena where a peak that is actually only a few pixels wide appears as a bright cross pattern centered at the peak with its legs running parallel to the rows and columns of the pixel array. Since the dc represents the average intensity of the original image, its information content is considered insignificant to pattern recognition. Therefore, this dc portion could be eliminated without detracting from the research. To do this, a high-pass filter (HPF) was generated using high resolution reduction photography (see Appendix A). The filter consisted of a circle 200 μm in diameter with a uniform attentuation of approximately 2000 times. A comparison of a *FT* of a square saved with and without the HPF is shown in Figure 9. It can be seen in Figure 9(b) that using the HPF enabled the CCD camera to resolve more diffracted orders of the *FT* image and eliminated the "blooming" that is evident near the center of Figure 9(a).

The schematic for taking the high pass filtered *FT* is shown in Figure 10. Mayo filtered his *FT* image by placing the HPF against the camera opening. Fig-



(*a*)

(*b*)

Figure 9. Comparison of FTs of a square: (a) No HPF, (b) Using HPF. Note the blooming (saturation) near the center in (a).

ure 10(a) shows that the CCD array was located in the focal plane of the Fourier transforming lens, and due to the camera body, the HPF was actually placed approximately 7–10 mm in front of the focal plane. This configuration made the filter difficult to center, and since the HPF was not in the focal plane of the FT lens, the desired blocking affect was not attained. A more accurate method of filtering is shown in Figure 10(b). The HPF was placed in the focal plane of the FT lens (L_1) and the focal plane of L_1 was imaged onto the CCD camera using a second lens L_2.

Two techniques were discovered that were very helpful in ensuring that the FT was properly setup. First, to ensure that the CCD camera was in the imaging plane of L_2, a multiple geometric object template was used as an input image. Due to the FT shifting property (see Chapter 2), objects located at different positions will still have dc terms located at the exact same position in the Fourier plane. Therefore, the distance between the imaging lens L_2 and the CCD camera can be adjusted until the dc terms overlap. Second, to ensure that the HPF was located precisely in the focal plane of L_1, the FT shift theorem was again exploited. Movement of a single object in the input plane will not change the location of the FT with respect to the HPF. Therefore, the distance of the HPF with respect to L_1 was adjusted until the dc and the HPF remained in the same location when the input object is moved. With these two techniques properly executed, the system was aligned and ready to record Fourier transforms.

Since a grey scale capable SLM was not available, we recorded FT images directly on 35 mm high contrast slide film. Polaroid's HC-135 instant slide film was found to give the best results. To make the 35 mm slides, a moveable mirror was placed between L_2 and the CCD camera (see Figure 10(b)). This mirror allowed switching between the CCD and the 35 mm camera. The mirror diverts the FT image to a single lens reflex (SLR) camera body (with no camera lens). The average power of the FT image at the SLR camera was measured to ensure that it was at eye safe levels. A half a milliwatt or less is considered eye safe. The level used in this

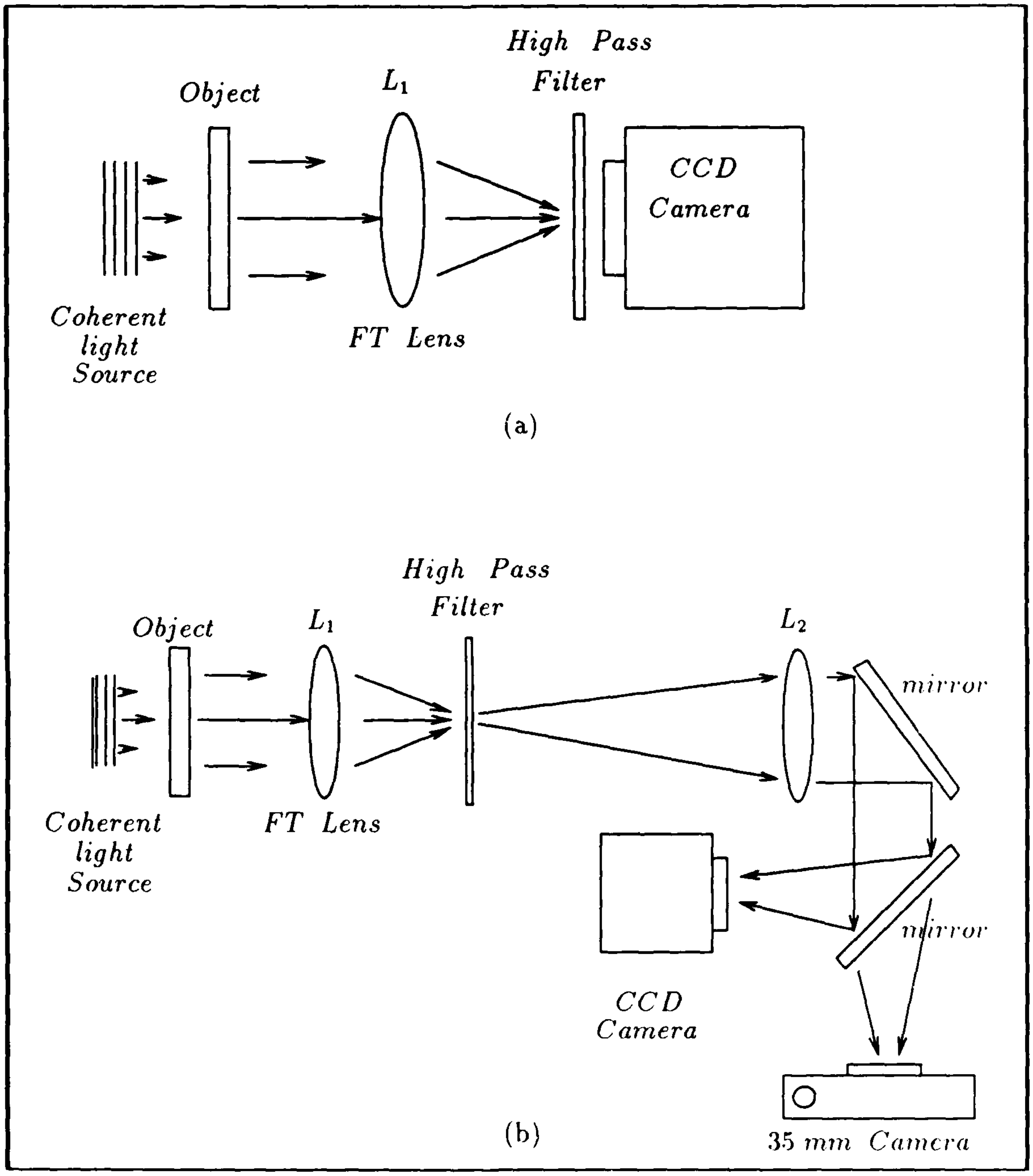


Figure 10. Optical Fourier transforming setup where (a) was Mayo's method and (b) was used in this research. Note: The HPF (dc-block) is in the back focal plane of L_1 to remove low frequency components.

setup was in the tens of microwatts. While sighting through the camera's viewfinder, the multiple object technique mentioned above was used to focus the $|FT|^2$ onto the film plane of the camera. The SLR camera's automatic mode was used only to estimate the shutter speed needed for a correct exposure since the automatic shutter tended to produce overexposed slides. Therefore, the camera's manual mode was used, bracketing two or three shutter speeds above the automatic shutter setting.

As with the binary templates, the Fourier transforms had to be digitally saved for later processing on the VAX. This was accomplished by turning the mirror shown in Figure 10(b) so the *FT* was incident on the CCD camera array and then "grabbing" the image in an identical manner as explained earlier.

The next step in the optical implementation is transforming the recorded Fourier transforms to the $f_{\ln r} - f_\theta$ coordinate transform space. This transformation is the subject of the following section.

3.4 Polar-Logarithmic Coordinate Transformation Using a CGH

3.4.1 Introduction This section discusses the optical implementation of a computer generated interferogram as performed in this thesis research. The CGH was used to coordinate transform the Fourier transform intensity patterns discussed in the previous section into $f_{\ln r} - f_\theta$ feature spaces. Detailed procedures and mathematical formulations used to solve for the interferogram and to produce the optically implemented version of the CGH can be found in Appendices A and B.

3.4.2 Optical Implementation The researchers' first attempt at optically implementing the $f_{\ln r} - f_\theta$ feature space was modeled after the setup used by Mayo as shown in Figure 11 [12:22]. The input image ($|FT|^2$) was placed directly against the CGH "in" plane P_1. Obviously, this is impossible. Since both the CGH and the input image (a Polaroid slide with holder) have physical dimensions, one cannot overlap them and still have both in the exact same plane. Additionally, since the center of the $|FT|^2$ must be aligned precisely with the center of the CGH for a symmetrical transformation, butting the CGH and input up against one another failed due to the lack of an accurate alignment procedure.

To overcome the in-the-same-plane problem, the $|FT|^2$ needed to be imaged onto the CGH. To overcome the alignment problem, the $|FT|^2$ slide needed to be mounted in an xyz-adjustable holder. The schematic shown in Figure 12 details the optical setup used to solve the problems outlined above.

The two-lens imaging system comprised of L_1 and L_2 accomplished three essential functions in this setup. First, L_1 and L_2 were separated by a distance equal to $f_1 + f_2$. This produced the plane wave coming out of L_2 that was required to illuminate the CGH, since only a real input could be coordinate transformed. Second, the ratio of f_2 to f_1 (750mm to 250mm) gave an effective magnification of three times. This magnification was required to overcome a space-bandwidth product (resolution) problem which will be discussed in more detail later. Finally, L_1 and L_2 provided the means of imaging the input directly onto the plane of the CGH.

With the $|FT|^2$ imaged onto the CGH, the combination of the binary interference pattern of the CGH and the Fourier transforming lens L_3 performs the coordinate transform function. The $f_{\ln r} - f_\theta$ CT can then be viewed at the back focal plane ($f_3 = 300mm$) of L_3. To view the CTs, the CCD camera was again used. Figure 13 shows a photograph of an optically produced CT of a small square.

3.4.3 Alignment and Focusing As mentioned in the previous section, aligning the input image's center with the center of the CGH was a problem. The following procedure was used to solve this problem. With the input (35mm slide mounted in slide holder) inserted in a Newport Corporation LP-2 lens holder, which gives xyz translation adjustment, the CT was viewed in real-time on a monochrome computer monitor. The AT&T TARGA framegrabber was again being used to grab and display the image incident on the CCD pixel array. As the xy adjustments of the LP-2 were

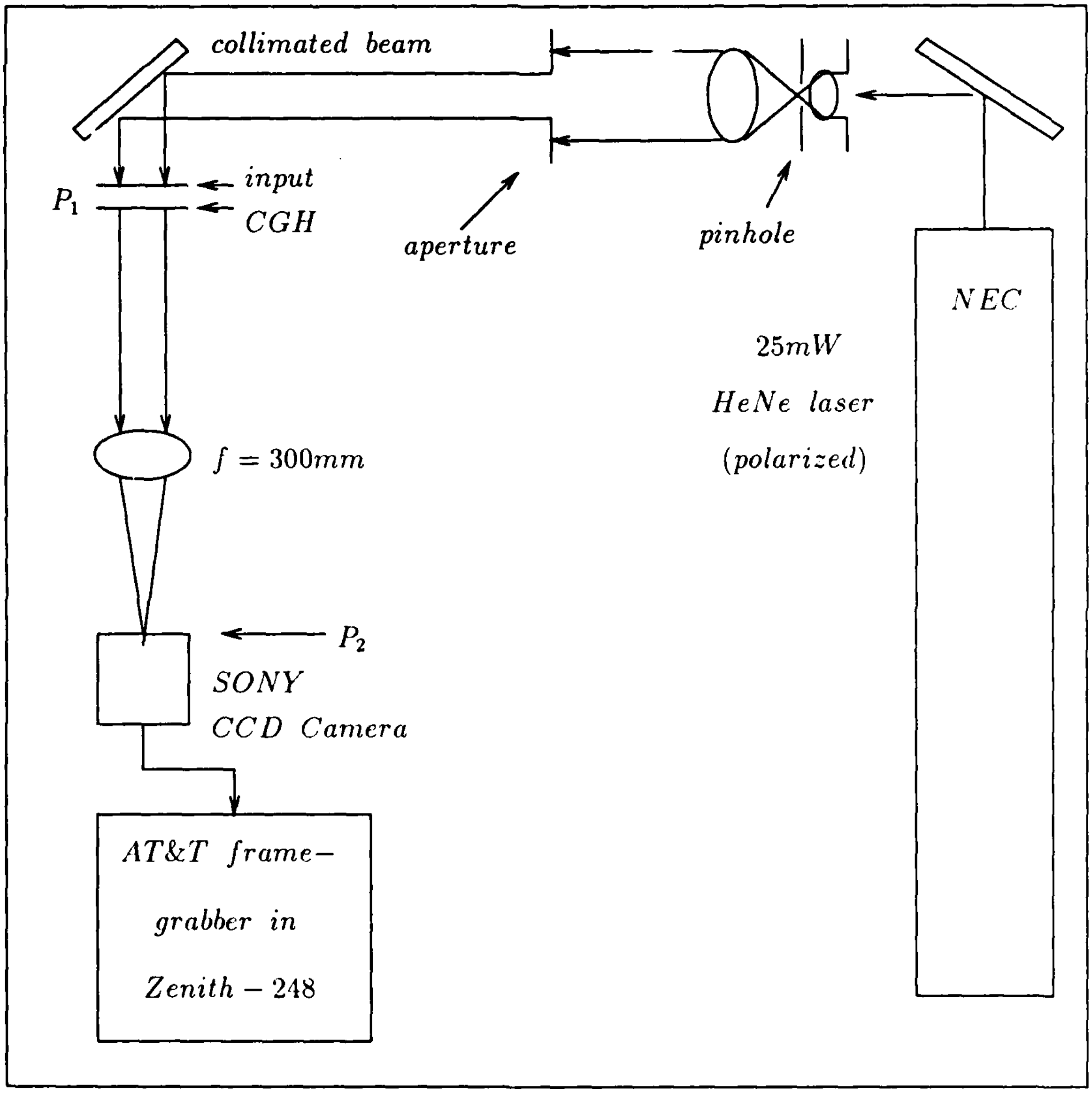


Figure 11. Mayo's $f_{\ln r} - f_\theta$ feature space setup. Note that the input is butted against the CGH [12:22].

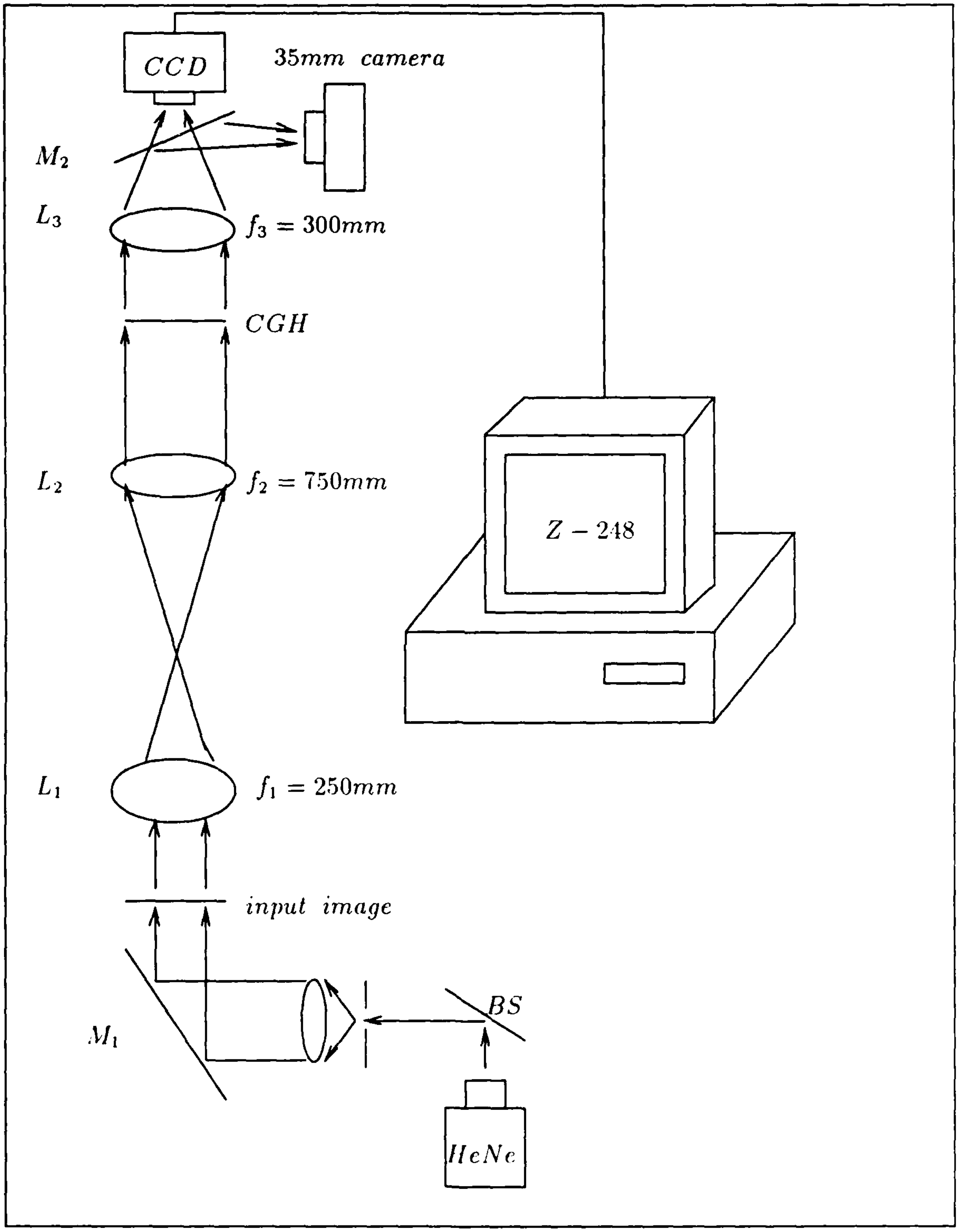
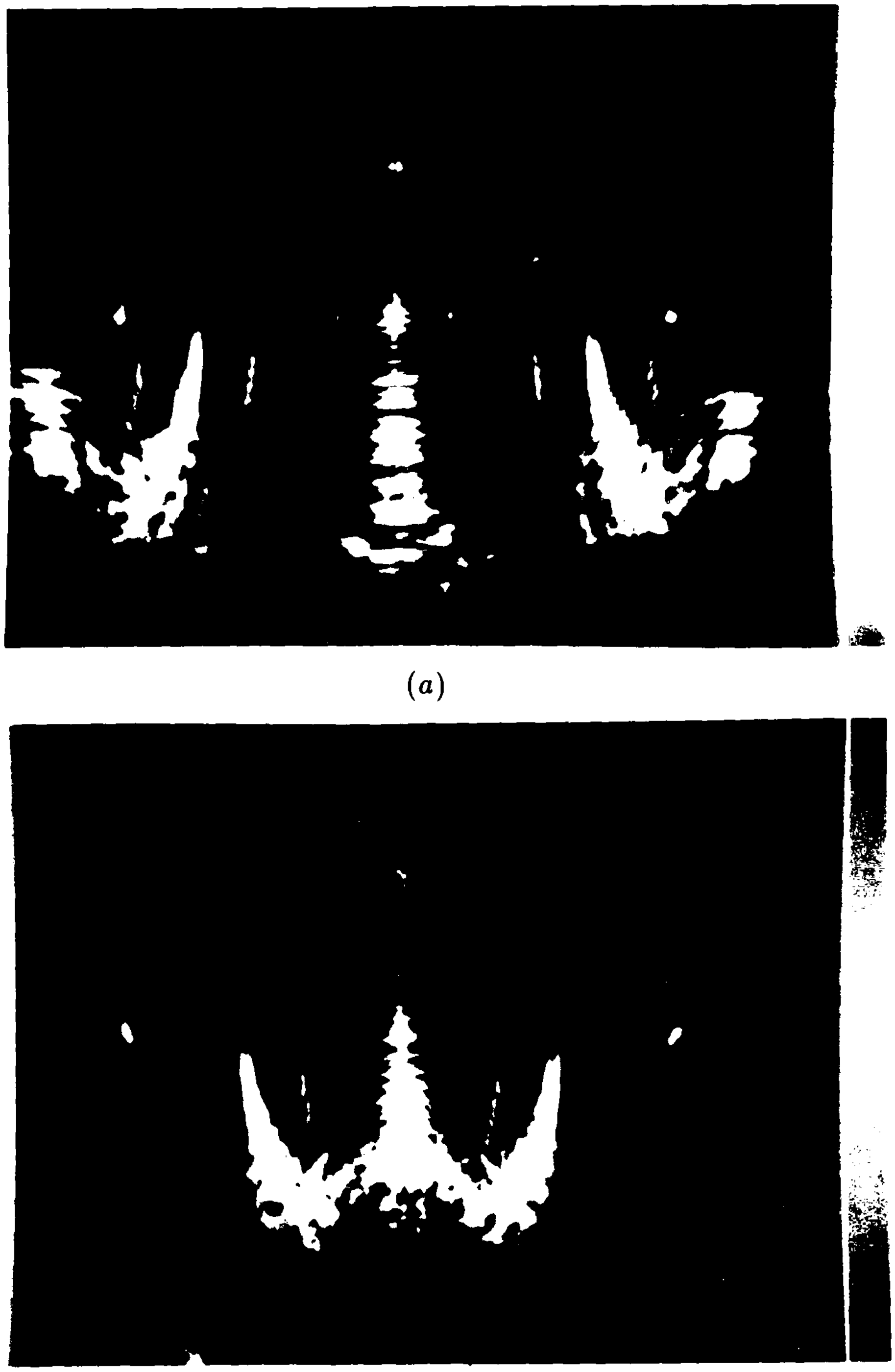


Figure 12. Optical setup for $f_{\ln r} - f_\theta$ CT. Note the addition of the imaging lenses L_1 and L_2 as an improvement over Mayo's design.

Figure 13. Photograph of the $\ln r - \theta$ CT of a small square. Note the symmetry and similarity in the heights of the peaks.

varied, the image of the CT was observed to drastically change shape. If the input image was too high with respect to the center-to-center alignment, the CT would fold upward. Similarly, when the input image was too low, the CT would fold downward. Finally, a left or right shift off of the center-to-center alignment would twist the CT as if it were being modulated by a sine wave. Figures 14 and 15 show photographs of the out-of-alignment patterns just discussed, as well as, a photograph of a correctly aligned CT.

Focusing the coordinate transforming system was a two-step process in itself. In the first step, the CCD camera was positioned in place of the CGH and used to view the magnified image of the $|FT|^2$ input. The z-axis adjustment of the LP-2 lens holder was then varied until a sharply focused image was observed on the monitor.



(*a*)

(*b*)

Figure 14. Out-of-Alignment CTs (a) image too high, and (b) image too low. Note: (a) is folded upward and (b) is folded downward.

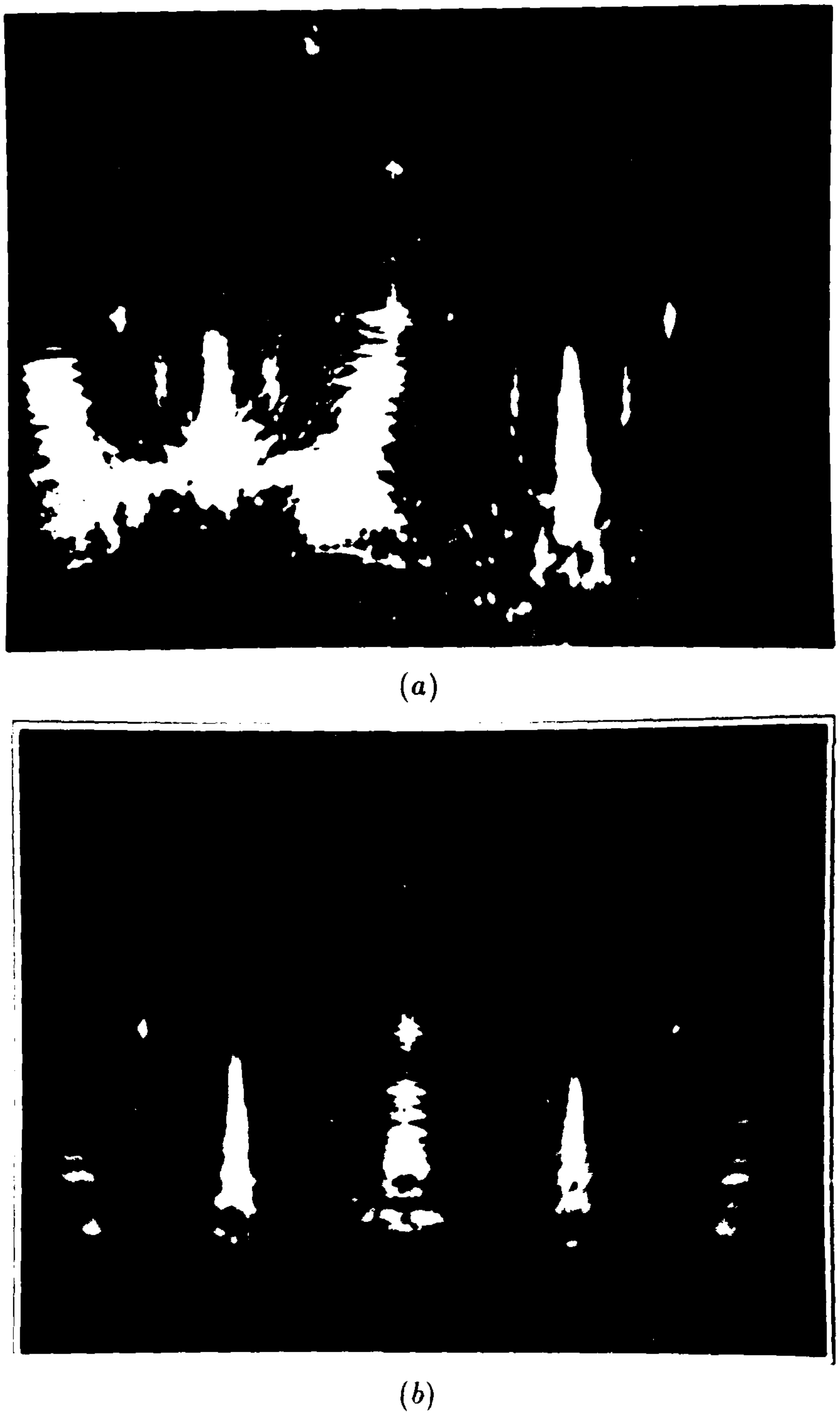


Figure 15. Alignment of CTs where (a) image was shifted left, and (b) image correctly aligned. Note the bending in (a) and the symmetry in (b).

In the second step, the CGH was returned to its initial position, the input slide was removed from LP-2, and the CCD was returned to the back focal plane of L_3. Now, to ensure that the CCD was in the focal plane of F_3 (which meant that the CT was in focus), the z-axis position of the CCD camera was varied while the monitor was observed. Since the input had been removed, the CGH (which is square) was being completely illuminated and the CT observed was that of a square. When the square's CT was in focus, all of its peaks were of equal height (see Figure 13). If the CCD is out of the focal plane, the heights of the peaks will not be uniform. Figure 16 shows photographs of out-of-focus CTs of a square.

3.4.4 Space-Bandwidth Product As mentioned earlier, the FTs had to be magnified by a factor of three to overcome a SBWP problem. The initial two-lens imaging setup attempted used two 125mm focal length lenses. This provided the necessary plane wave generation, but only accomplished a one-to-one imaging onto the CGH. Thus, because the $|FT|^2$ on the Polaroid slide film was small (approximately 2-3 mm), the image on the CGH was small. Apparently, because of the SBWP associated with the interferogram (35 lines/mm, 10mm x 10mm), sufficient energy from the $|FT|^2$ was not being diffracted into the $f_{\ln r} - f_\theta$ CT feature space. Similar results were being observed with digital simulations and it was not until the $|FT|^2$ was magnified that the expected CT was observed. A magnification factor of three was chosen because it provided a good CT without causing too many of the lower orders of the FT to be cut off by the physical extent of the CGH. Figures 17 and 18 are a series of four photographs which explicitly show the SBWP problem. The photographs show the CTs of circular apertures of varying sizes that were imaged onto the CGH. It can be seen that the resolution degrades as the aperture size is reduced. At an aperture size of one millimeter, the CT has completely deteriorated.

3.4.5 Linearity One last aspect of the CGH that was explored was its linearity with respect to scale and rotation changes. Rotation linearity was inherent and

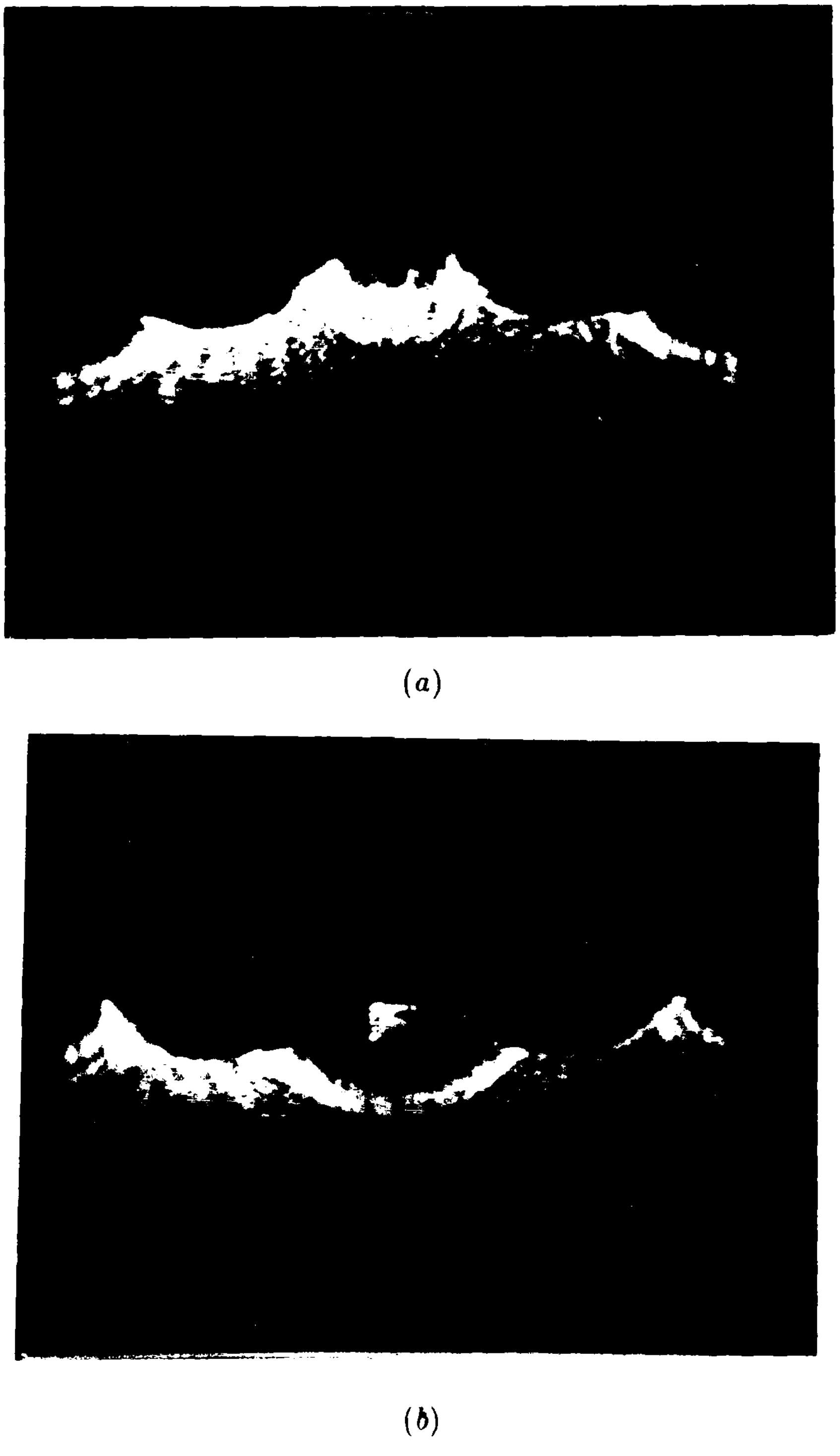
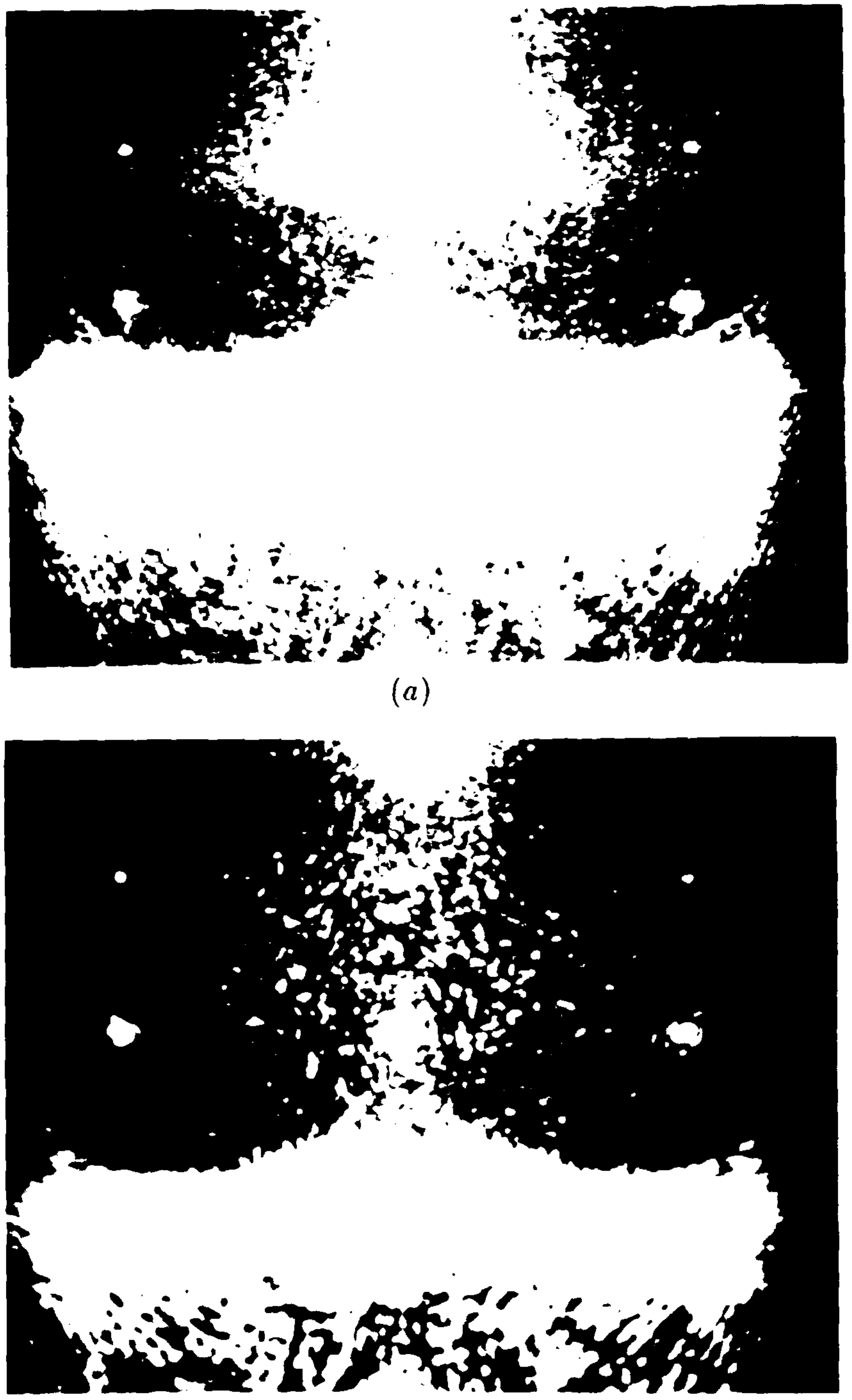


Figure 16. Out-of-focus CTs of a square (a) pre-focal plane, (b) post-focal plane. Note the higher inner peaks in (a) and the higher outer peaks in (b).



(a)

(b)

Figure 17. CTs of circular apertures where (a) is of an 8mm aperture and (b) is of a 4mm aperture. Note how the CT has shifted down in (b).

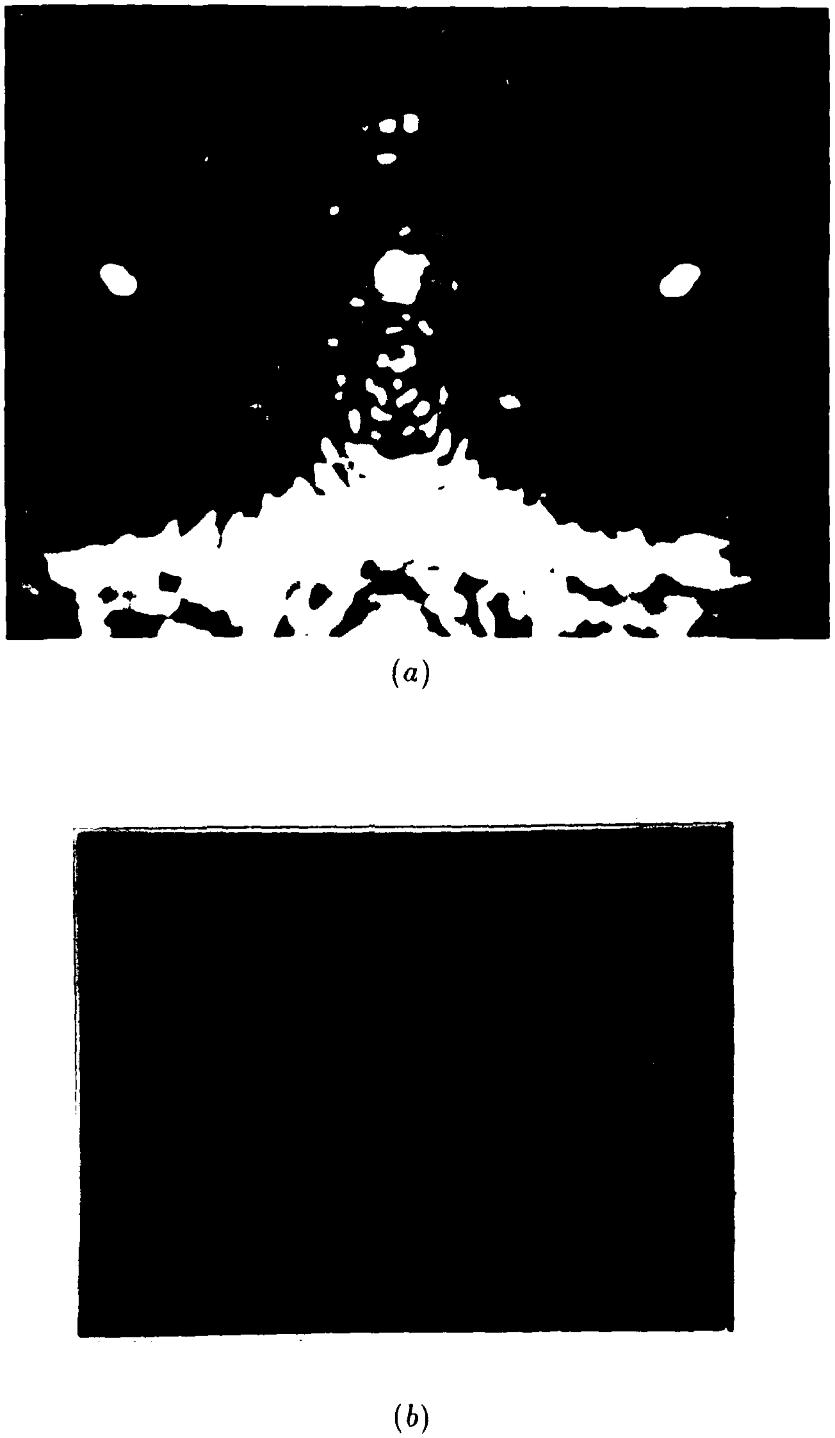


Figure 18. CTs of circular apertures where (a) is of a 2mm aperture and (b) is of a 1mm aperture. Note the deterioration in these CTs.

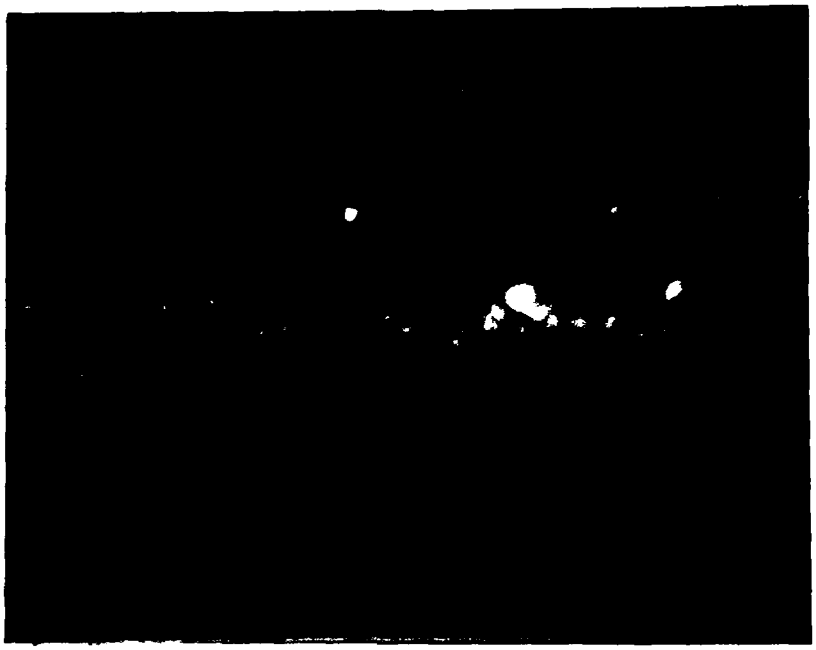


Figure 19. CT of square rotated 45 degrees. Note linear shift along horizontal (f_θ) axis with respect to Figure 13.

obvious (see Figure 19), but scale-change linearity needed to be verified. To do this, the optical setup shown in Figure 20 was used to test the CGH's response to various sized circular apertures.

A mechanical iris was used as a variable aperture source. The diameter of the aperture was varied from one millimeter to nine millimeters in one-millimeter increments. At each aperture setting, the location of the CT (a horizontal "line" for a circle) on the monitor was marked and recorded. Results of these measurements are recorded in Figure 21(a) and shown on the graph in Figure 21(b), where the CGH shift is shown to be nearly linear for aperture sizes of four millimeters and above. The theoretical shift values shown in Figure 21(a) are based on the $\ln(a)$ shift expected from Equation 5 and referenced to two millimeters. No value for a

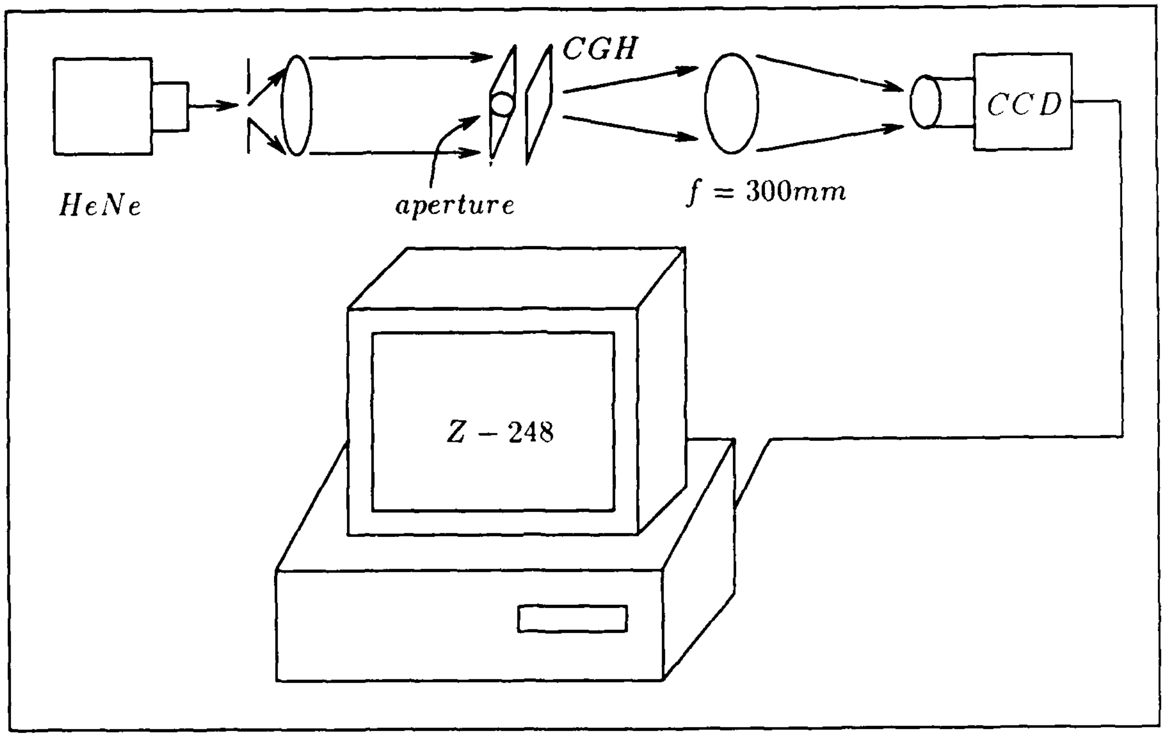
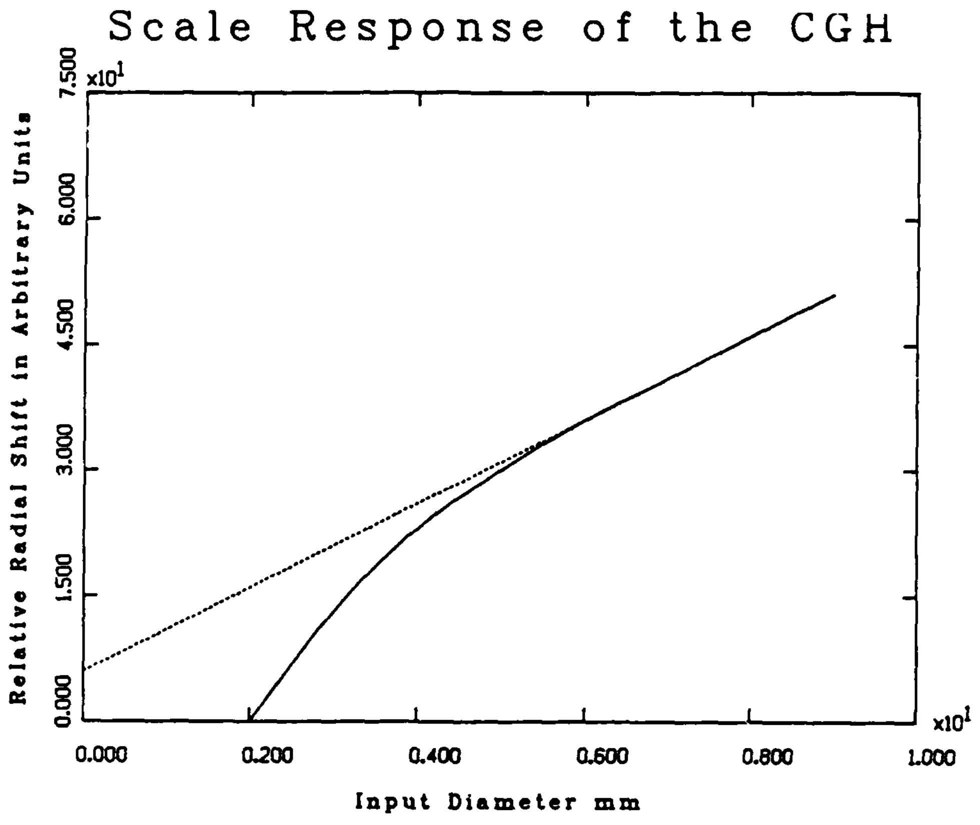


Figure 20. Optical setup to test linearity of $f_{\ln r} - f_\theta$ CGH.

Input Dia. (mm)	Relative Shift (mm)	Theoretical Shift (mm)	Error (%)
1	—	—	—
2	0	0	0
3	13	13.7	5.1
4	23	23.0	0
5	30	30.7	2.4
6	36	36.7	1.9
7	41	41.7	1.7
8	46	46.3	0.6
9	51	50.0	2.0

(*a*)



(*b*)

Figure 21. Table and Graph of aperture size versus distance from zero reference. (a) Table shows maximum error of 5.1%, (b) Graph shows linear region above 4mm. Note: The dotted line extrapolation shows the linear portion of the ln curve.

one millimeter aperture was possible because the CT had deteriorated. Thus, the two millimeter reading was used as the reference for relative shift. Notice that the largest percent error was only 5.1%, showing that the CGH does, in fact, give the expected $\ln(a)$ shift.

3.4.6 Data Storage As with the *FT*, two types of data needed to be recorded. First, the CCD camera, with the framegrabber, was used to save images to disk in the same manner as discussed earlier. Second, when mirror M_2 (see Figure 12 was rotated into the optical pathway, the CT was focused onto the filmplane of the 35mm camera. Focusing was accomplished by viewing the CT of a square through the camera's viewfinder, and then translating the camera until the peaks of the CT were of uniform height. The Polaroid HC-135 instant slide was again used to record the CTs.

Once all of the CTs were recorded, selected slides of the $|f_{\ln r} - f_\theta|^2$ feature space were used to optically generate Vander Lugt filters for correlation experiments. These procedures are the topic of the next section of this chapter.

3.5 Correlating The Feature Space

3.5.1 Vander Lugt Filtering In 1963, A. B. Vander Lugt proposed and demonstrated a coherent light technique for frequency-plane matched filtering. The beauty of his technique is that both the amplitude and the phase of a transfer function can be recorded on film as an amplitude function (or interferogram). Furthermore, this spatial frequency information can be physically compared to the spatial frequency information of an input image. Why is this technique so significant? In the spatial frequency domain, both amplitude and phase information are needed to completely characterize a signal. The amplitude frequency spectrum only describes part of the frequency response of a system. It is the the phase information that determines the position of these spatial frequencies in the impulse response of the optical system.

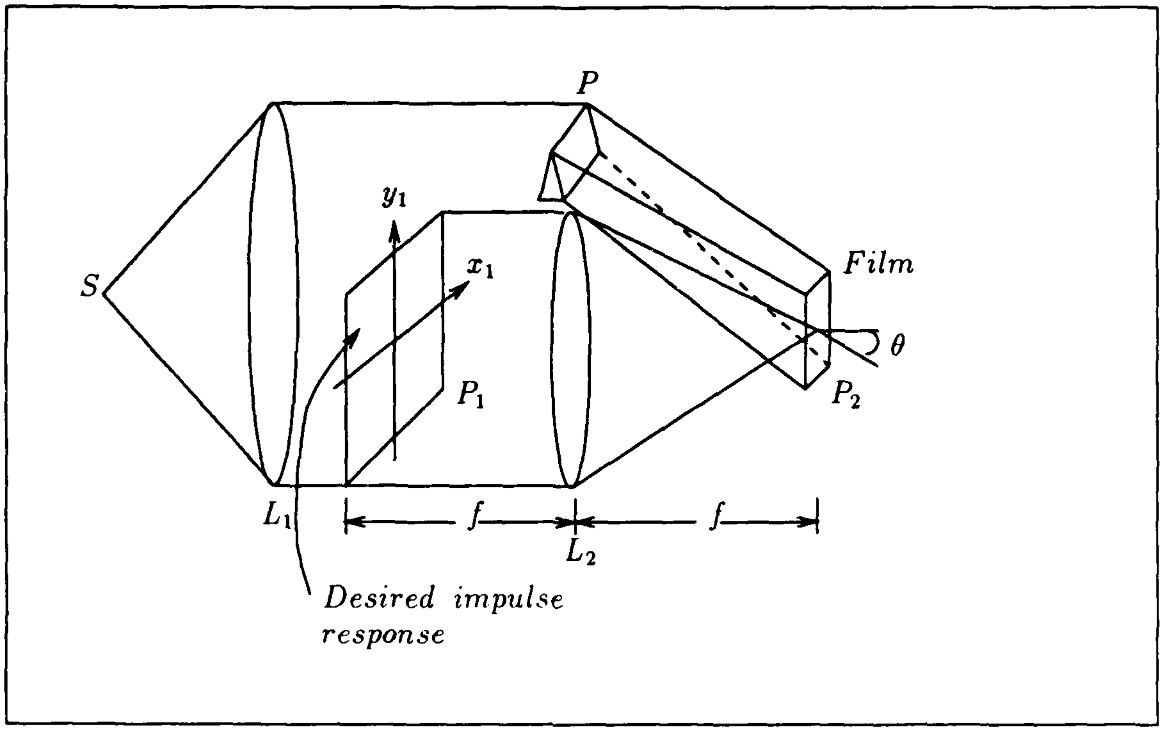


Figure 22. Optical Schematic for generating Vander Lugt filter [6:172] .

Figure 22 shows a typical schematic for generating a Vander Lugt filter. A transparency of a desired impulse response (h) for an optical system is illuminated with a collimated, coherent light source at plane P_1 . Lens L_2 , located one focal length (f) in front of the transparency, Fourier transforms the transparency with the resultant scaled transfer function ($\frac{1}{\lambda f}\mathbf{H}(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f})$) (note bold faced type indicates complex function) focused on a high resolution photographic plate at P_2 . While the film is exposed with the transfer function, the film is also being exposed with a plane wave at an angle θ with respect to the path of the illuminated impulse response. This plane wave $\mathbf{U}_r$ is commonly called the "reference beam", whereas the illumination of the transfer function is called the "object beam". Note that the path length difference between the object and reference beams must be within the coherence length of the light source to ensure an interference pattern between these beams is generated on the film plane.

The reference beam is described by

$$\mathbf{U}_r(x_2, y_2) = r_o \exp(\frac{-j2\pi y_2 \sin\theta}{\lambda}) \tag{13}$$

where r_o is the amplitude of the reference beam and the exponential term is the phase difference along the reference beam at the film plane P_2 due to the angle of incidence θ. The resultant intensity distribution (I) at P_2 is equal to the square of the sum of the amplitudes of the reference and object beams:

$$\begin{aligned} I(x_2, y_2) &= \left| r_o \exp(\frac{-j2\pi y_2 \sin\theta}{\lambda}) + \frac{1}{\lambda f}\mathbf{H}\left(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}\right) \right|^2 \\ &= r_o^2 + \frac{1}{\lambda^2 f^2}|\mathbf{H}|^2 + \frac{r_o}{\lambda f}\mathbf{H}\exp(j2\pi\alpha y_2) \\ &\quad + \frac{r_o}{\lambda f}\mathbf{H}^* \exp(-j2\pi\alpha y_2) \end{aligned} \tag{14}$$

where $\alpha = \sin\theta/\lambda$.

In order to demonstrate the amplitude and phase retention of the Vander Lugt interference technique, the transfer function's wavefront can be rewritten as an amplitude function multiplied by a phase function:

$$\mathbf{H}(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}) = A(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}) \exp\left[-j\psi\left(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}\right)\right] \tag{15}$$

Substituting Equation 15 into Equation 14 yields a very descriptive equation for the intensity distribution of the interference pattern at the film plane:

$$\begin{aligned} I(x_2, y_2) &= r_o^2 + \frac{1}{\lambda^2 f^2}A^2\left(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}\right) \\ &\quad + \frac{2r_o}{\lambda f}A\left(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}\right)\cos\left[2\pi\alpha y_2 - \psi\left(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}\right)\right] \end{aligned} \tag{16}$$

This equation shows that the interference pattern is the transfer function's amplitude and phase functions modulated onto a high frequency carrier.

Exposing this interference pattern to a photographic plate yields a transparency with an amplitude transmittance function $t(x_2, y_2)$ with a distribution

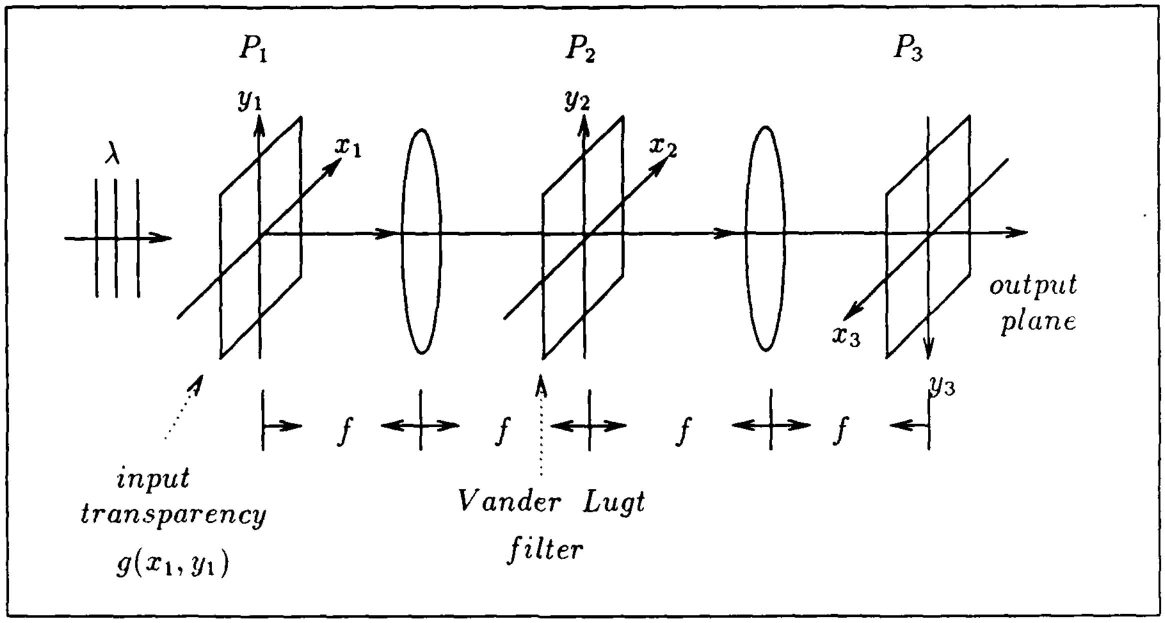


Figure 23. Optical schematic to generate optical correlation [6:167] .

proportional to intensity functions listed in Equations 14 and 16. By next removing the reference beam and the impulse response transparency at P_1 , this optical processing schematic can be modified to perform signal processing. The post-filter generation optical schematic is shown in Figure 23. In Figure 23, the optical system's impulse response transparency is replaced with a transparency of an input signal $g(x_1, y_1)$ to be processed (filtered) by the system. In addition, a Fourier transforming lens L_3 is added to the output of the Vander Lugt filter to convert the processed signal back to the space domain.

The input signal $g(x_1, y_1)$ is processed in the following manner. First, the input signal is Fourier tranformed onto the Vander Lugt filter at P_1 . This is equivalent to multipling the input signal's frequency information with the amplitude distribution of the filter. Therefore, the resultant field $\mathbf{U}_2$ transmitted through the filter is

$$\begin{aligned}\mathbf{U}_2 \;\propto\; & \frac{r_o^2\mathbf{G}}{\lambda f} + \frac{1}{\lambda^3 f^3}|\mathbf{H}|^2\mathbf{G} + \frac{r_o}{\lambda^2 f^2}\mathbf{H}\mathbf{G}\exp(j2\pi\alpha y_2) \\ & + \frac{r_o}{\lambda^2 f^2}\mathbf{H}^*\mathbf{G}\exp(-j2\pi\alpha y_2) \end{aligned} \tag{17}$$

Next, $\mathbf{U}_2$ is Fourier transformed with a resultant field

$$\begin{aligned}\mathbf{U}_3(x_3,y_3) \;\propto\; & r_o^2\mathbf{g}(x_3,y_3) + \frac{1}{\lambda^2 f^2}[\mathbf{h}(x_3,y_3) * \mathbf{h}^*(-x_3,-y_3) * \mathbf{g}(x_3,y_3)] \\ & + \frac{r_o}{\lambda f}[\mathbf{h}(x_3,y_3) * \mathbf{g}(x_3,y_3) * \delta(x_3, y_3 + \alpha\lambda f)] \\ & + \frac{r_o}{\lambda f}[\mathbf{h}^*(-x_3,-y_3) * \mathbf{g}(x_3,y_3) * \delta(x_3, y_3 - \alpha\lambda f)] \end{aligned} \tag{18}$$

The third and fourth terms of Equation 18 are the most important results of this optical system. The third term yields the complex convolution of the input image with the desired impulse response centered at coordinates $(0, -\alpha\lambda f)$ in plane P_3. The fourth term yields the crosscorrelation between the input image and the impulse response centered at the coordinates $(0, \alpha\lambda f)$ in P_3.

The Vander Lugt filter's inherent ability to generate two-dimensional cross correlations at light speed indicates that this technique may prove very useful in future attempts at building a near-real-time pattern recognition machine. For crosscorrelations between the feature spaces of a target template and an image, the template feature space is the desired impulse response and the image feature space is the input signal to be processed. For this thesis, Vander Lugt filters were generated for the Fourier transform $f_{\ln r} - f_\theta$ feature space of target templates and then used to correlate the feature spaces of test images. To create Vander Lugt filters optically, holographic techniques requiring chemical developing, fixing, and bleaching are normally used. In this thesis, however, a thermoplastic holographic medium was used to make the filters.

3.5.2 Thermoplastic Holograms Several types of media are available for recording Vander Lugt filter interference patterns. The most common filter media are the

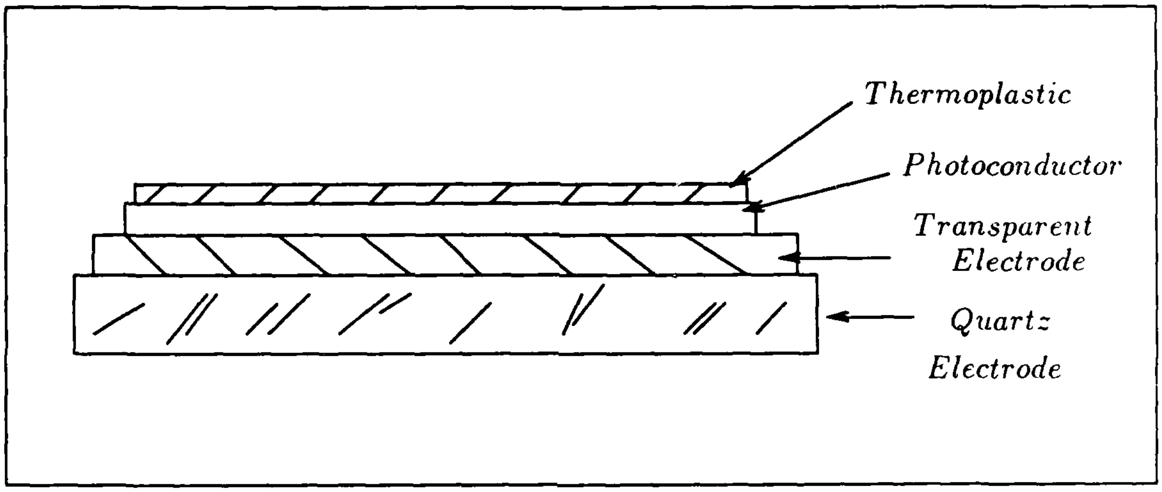
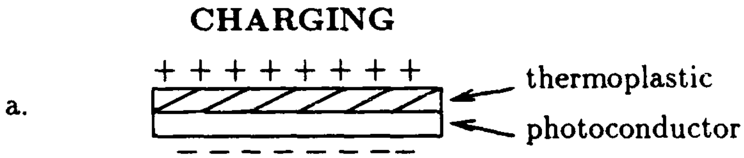


Figure 24. Cross section of thermoplastic film. Note: The thermoplastic layer is only $0.7\mu m$ thick [15] .

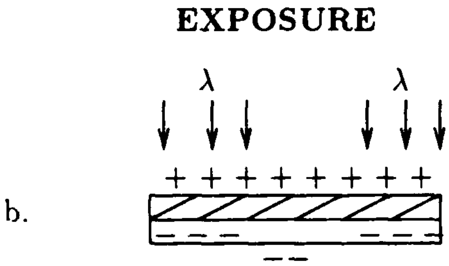
holograhic and high resolution photograhic films. But these materials require wet chemical processing for development. This research was conducted using a Newport HC-300 thermoplastic holographic recording device. This device is convenient in that it can generate a hologram, or for this research a matched-filter, in approximately 60 seconds without wet chemical processing. Additionally, the film is reuseable up to 1000 times. The HC-300 consists of a camera for holding and processing the thermoplastic media, a shutter system, a power sensor for automatic shuttering, and a system controlling unit.

The thermoplastic film (plate) is a multilayered media that has been deposited on a quartz substrate (see Figure 24). The film consists of three layers: thermoplastic, photoconductor, and transparent electrical conductor. Generating a thermoplastic hologram is a four step process (see Figure 25).

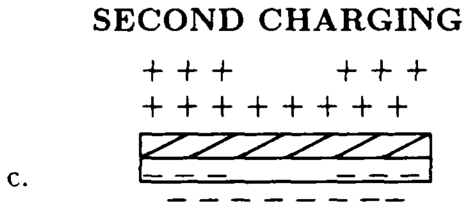
First, a uniform charge is deposited on the film using a moveable coronatron located inside the camera. As shown in Figure 26,the coronatron consists of a wire anode located on the central axis of a slotted cylinder. The anode voltage is raised



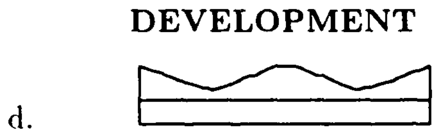
A uniform charge is deposited on the thermoplastic and photoconductor.



Exposure to laser fringe pattern redistributes the changes the photoconductor.



Recharge increases the electric field across the exposed area of the thermoplastic plate.



Heating of the thermoplastic causes permanent deformation. The irregularities will diffract light to recreate the original image.

Figure 25. Four step development process for generating a thermoplastic hologram: charging, exposure, recharging, and heating [15] .

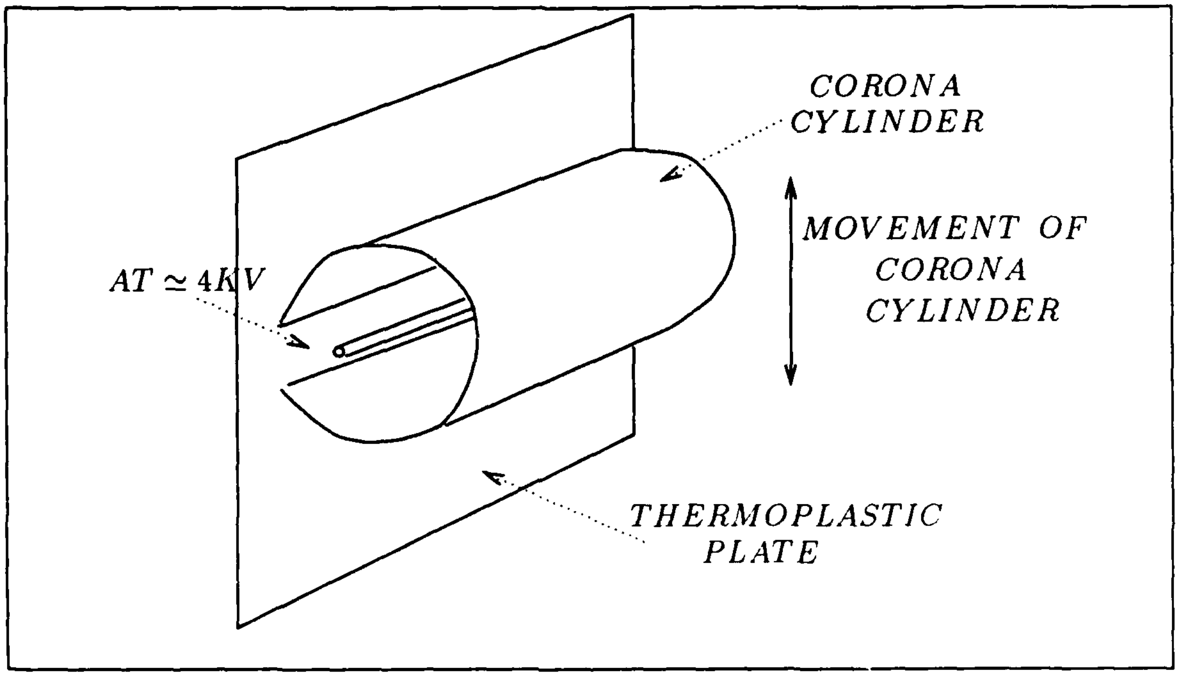


Figure 26. Coronatron charging a thermoplastic plate [15] .

to 4 kilovolts while the cylinder and the thermoplastic plate are grounded. The high anode voltage ionizes the air allowing electrons to collect on the anode while ions are collected on the cylinder. As the coronatron moves across the thermoplastic plate, some of the newly generated ions slip through the cylinder's slot (which is positioned next to the plate) and charge the medium.

The second step is to expose the plate with an object and reference beam. The plate's photoconductor layer allows negative charges to pass from the transparent conductor toward the thermoplastic. It should be noted that even though charge has migrated, a uniform electric field still exists across the plate.

During the third step, the plate is re-ionized by the coronatron. Since the exposure has terminated, the second charging allows more charge to build up on the exposed areas of the plate.

Finally, a current is passed through the transparent electrode of the plate to create heat for softening the thermoplastic. While the thermoplastic is soft, the plate's areas that were previously illuminated will be compressed by the increased electrostatic field produced by the plate's second charging. The plate areas that were exposed less will tend to bulge out. The result is a phase-relief hologram with a diffraction efficiency up to 10 percent.

3.5.3 Experimental Setup for Correlations Figure 27 shows the optical configuration used to demonstrate optical correlations of the Fourier transform $\ln r - \theta$ (FLRT) feature space. The first step was to generate a Vander Lugt filter of the FLRT feature space of a target template. The coherent light source was a 35 milliwatt helium-neon laser (632.8 nm wavelength). The laser's coherence length was measured (using the Michaelson interferometer technique) to be approximately 38 cm. The reference and object beam pathlengths were adjusted to within 0.5 cm. As recommended in Newport's HC-300 holographic recording device operating manual, the object beam was placed at an angle of 31 degrees with respect to the reference beam, and the object beam was attenuated with neutral density filters down to a 1:10 power ratio with respect to the reference beam. The target and test image's FLRT feature spaces were previously generated and recorded on 35 mm Polaroid Polagraph high contrast black and white slide film using the same techniques described for FTs and CTs. Using the HC-300 in automatic mode, a thermoplastic plate was exposed and developed as a matched filter for the target template's FLRT feature space. The diffraction efficiency of the filters ranged from 1-3 percent, which is lower than the 10 percent value published by Newport Corporation [15].

Next, the Vander Lugt filter was used to correlate test image FLRT feature spaces with the target's FLRT feature spaces. Note that since the *HC-300* holographic camera also acts as the filter's holder after development, the filter was not moved during the correlation experiment. To generate the correlated feature spaces, the feature space slide used to create the matched filter was replaced with a feature space slide of a test image. The reference beam was blocked from the matched fil-

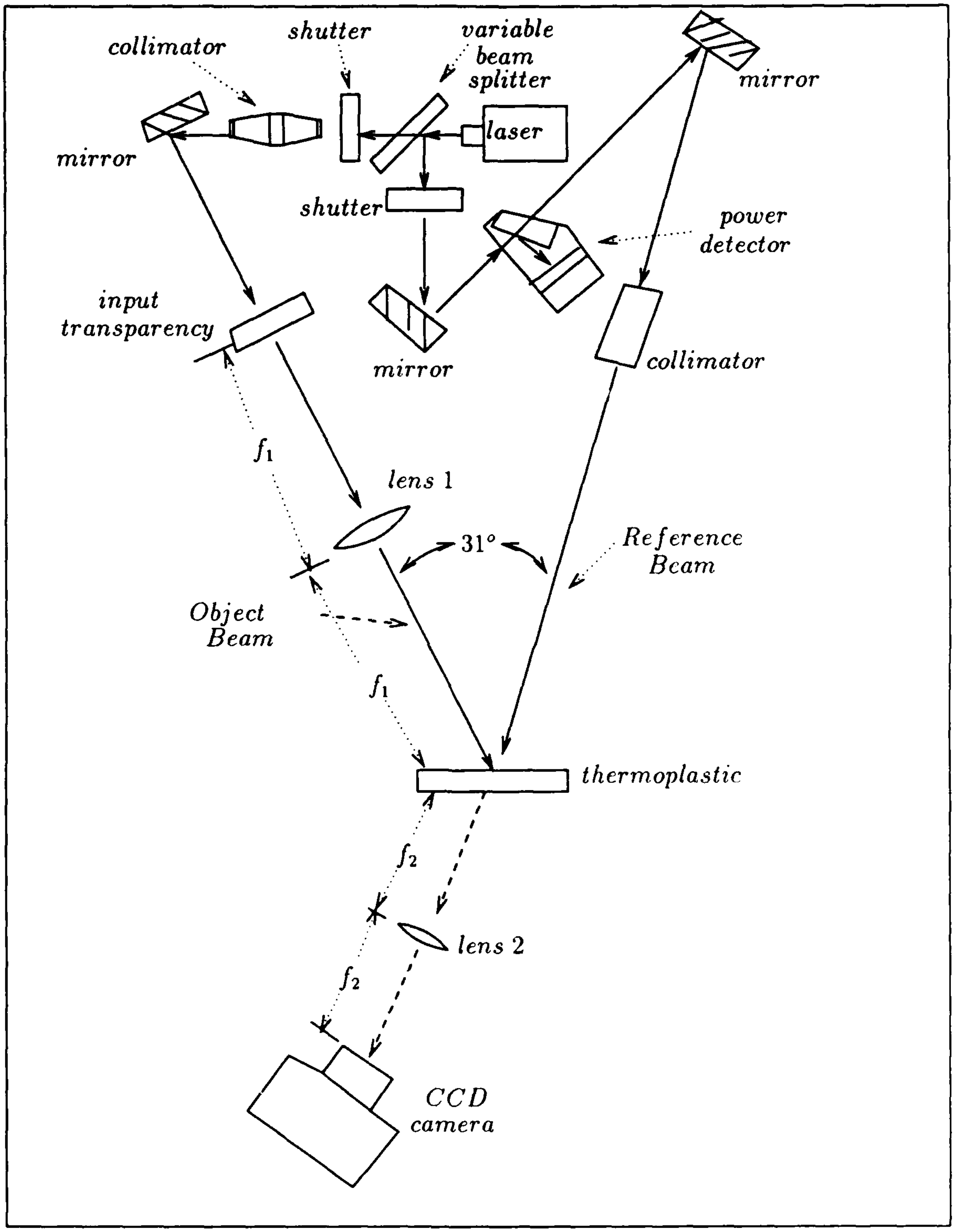


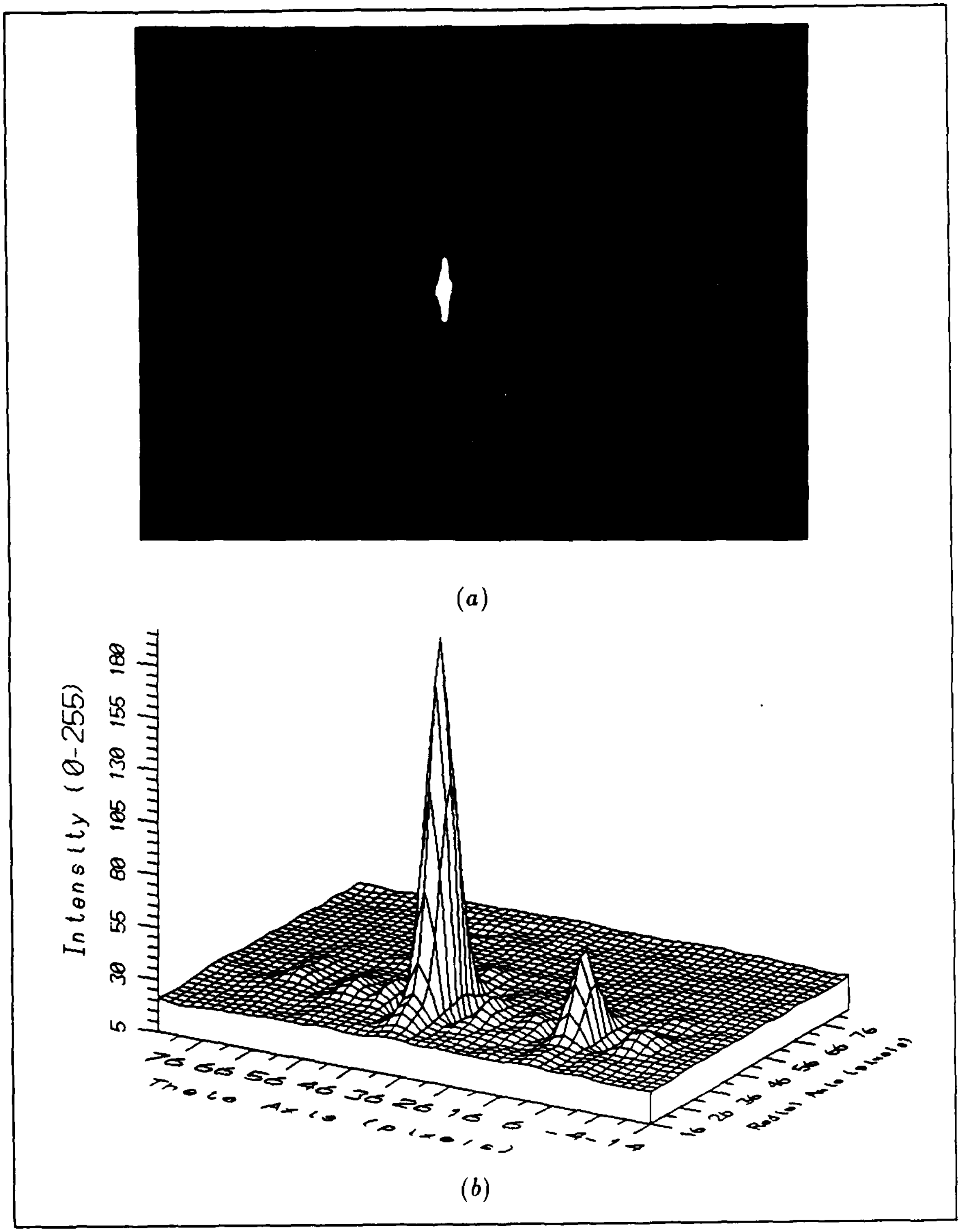
Figure 27. Optical setup for correlating the FLRT feature space.

ter, and the object beam illuminated the test image slide. The object beam travels through the transparency and a Fourier transforming lens to the matched filter. At the matched filter, the beam is diffracted into three separate beams. The beam that contains the correlation information, located 31 degrees away from the object beam's path, travels along the extended reference beam path. The correlation information passes through a Fourier transform lens and is then imaged onto a CCD camera where the correlation is viewed and digitally recorded. Figure 28(a) shows the optically generated autocorrelation function of a square. In Figure 28(b), the correlation function of (a) has been digitally profiled into the 3-D surface plot shown.

Since correlation peak location was essential for determining the scale and rotation of a target, alignment of the feature space transparencies was very important. Figure 29 shows that the optically generated feature spaces all have an artifact in common. When this artifact was enlarged, it was found to be the FT of the image that was originally incident on the computer generated hologram. Since the magnitude of the Fourier transform is position invariant, this artifact was located at the same exact position of every FLRT feature space. Therefore, this artifact was used to ensure that each test feature space was aligned at the same position of the template feature space used to generate the matched filter. Note that in Miazza's architecture (see Figure 5), image alignment is not a factor because spatial light modulators are used instead of transparencies.

3.6 Digital Simulations

To ensure the validity of the optical generation and correlation of the FLRT feature space, the optical coordinate transforms were simulated and correlated digitally. A digital model of a $\ln r - \theta$, phase-only filter simulated the optical coordinate transform. Note that the optical architecture used in this thesis used a computer generated hologram (CGH). The CGH is actually a binarized interferogram of the phase-only filter's output wavefront interfered with a plane wave reference beam.



(a)

(b)

Figure 28. Optical autocorrelation of the CT of a square. Note: (a) is an actual photograph and (b) is its 3-D surface plot.

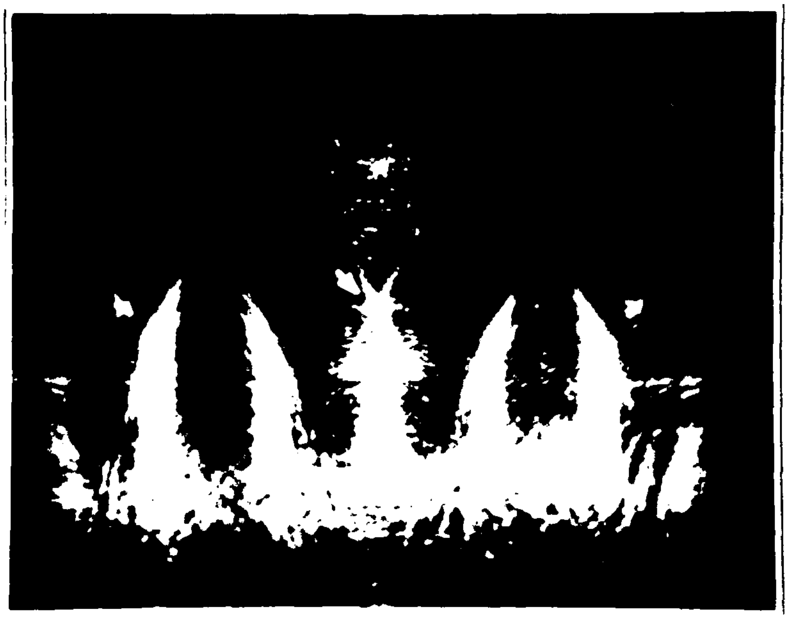


Figure 29. CGH artifact where arrow points to artifact used to align the feature space for correlating.

The CGH produces a coordinate transform at the first diffracted order of its output; whereas, the phase-only filter's transform is located on the optical axis.

"Hologram_menu", a software package written by Marvin Hill, was adapted to digitally simulate optical $\ln r - \theta$ coordinate transformations. Appendix E presents pertinent portions of Hologram_menu's code used in this thesis. Initially, computer generated Fourier transforms of the 256 by 256 pixel test images were used with Hologram_menu to generate the FLRT feature space. However, Hologram_menu's fast Fourier transform routine created transforms that were too small for the digital phase-only filter to generate a recognizable coordinate transform. Consequently, optical Fourier transformed images generated by the procedures described earlier in this chapter were recorded digitally using a framegrabber (see Appendix D) and then used as the input for Hologram_menu. A FLRT feature space, generated by Hologram_menu, of a small square rotated by 45 degrees is shown in Figure 30. Note that this feature space has an additional coordinate transform of a square draped across the top of the expected result. This additional energy was caused by the limiting aperture of the input image and the digital phase-only filter. The optically generated Fourier transformed images contained some energy at the edges of the 256 by 256 pixel (square image) array; and therefore, this energy was also mapped by the phase function. This additional intensity pattern was observed in all digitally coordinate transformed Fourier transforms. This redundant intensity pattern posed additional problems in that digital correlations of feature spaces were dominated by the pattern's autocorrelation peak. To alleviate this problem, the optically generated Fourier transforms were multiplied with a Hamming function

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1 \tag{19}$$

The result of preprocessing the optical Fourier transform with the Hamming function is shown in Figure 30 (b).

Kobel and Martin's computer program "Executive", was used to generate linear correlations of the feature spaces created with Hologram_menu. A crosscorrelation of the FLRT feature spaces of a large square with a square scaled by a half and rotated by 45 degrees is shown in Figure 31.

3.7 Summary

In this chapter, optical implementation and digital simulation of portions of the Kobel-Martin-Horev algorithm were discussed. Section 3.3 detailed the Fourier transforming property of lenses and explained how a dc-block was used to increase resolution in the optical system. This section also related two techniques that provided precise alignment of the dc-block and focusing of the CCD camera.

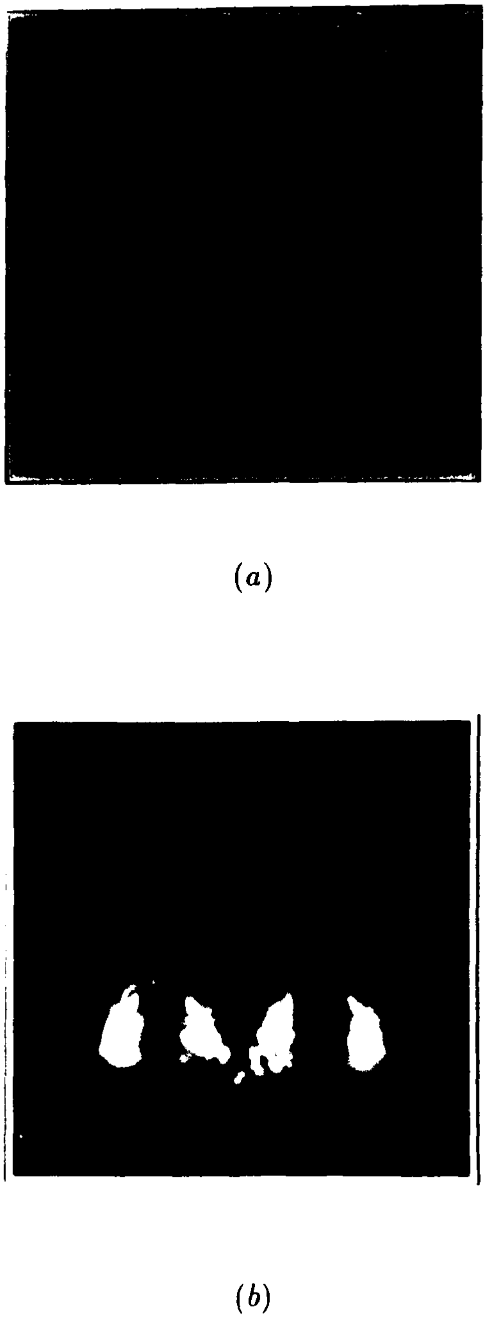
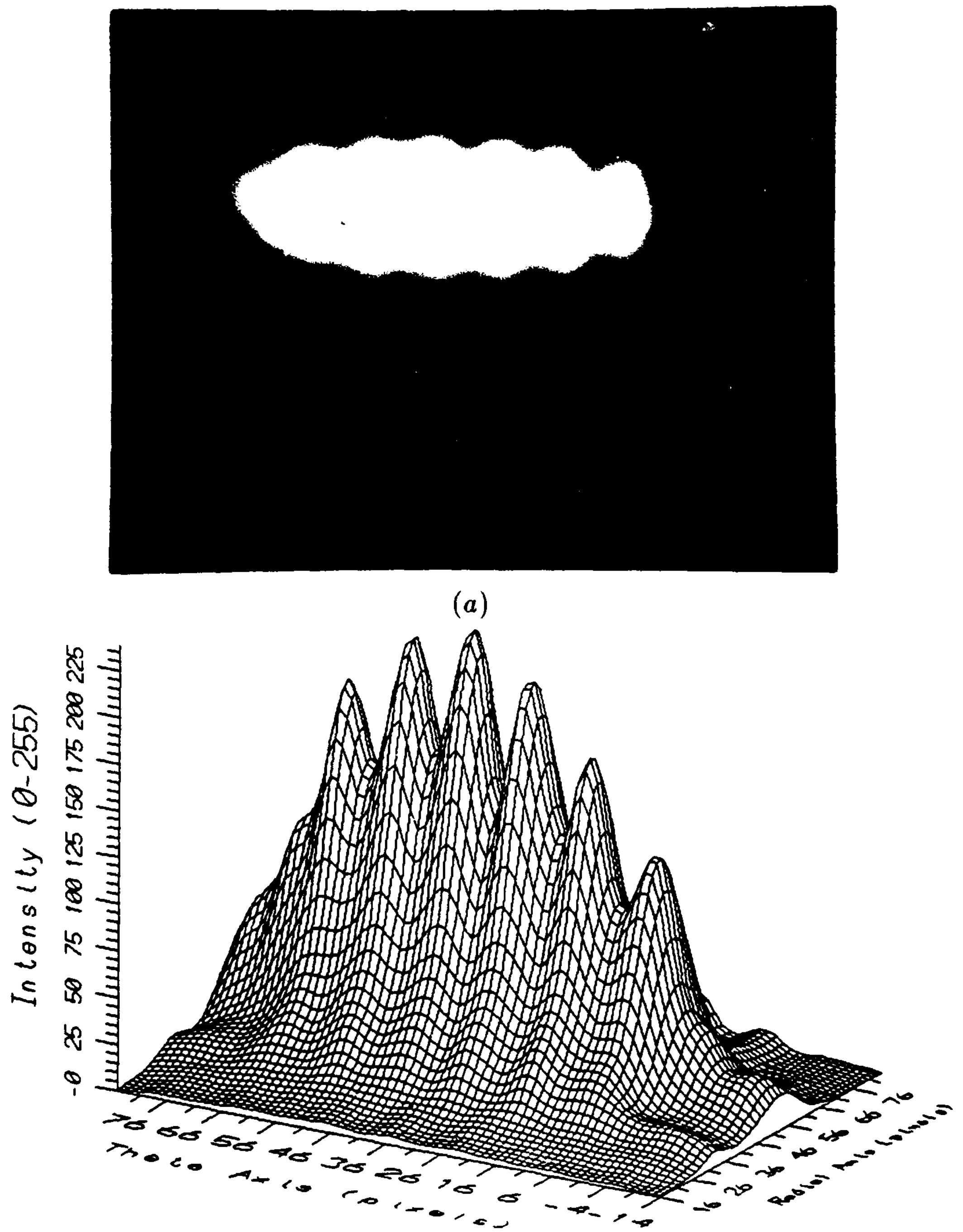


Figure 30. Digitally generated optical $f_{\ln r} - f_\theta$ transform of the optical FT of a small square rotated by 45° where (b) is same as (a) but preprocessed with a Hamming function.



(a)

(b)

Figure 31. Digital correlation of FLRT feature spaces of square with square scaled by a half and rotated 45° where(a) is a photograph of the intensity distribution and (b) is its 3-D surface plot.

Section 3.4 discussed the use of a CGH to perform a $\ln r - \theta$ CT. Several important improvements over previous attempts at optical implementation are detailed. These include imaging onto the CGH instead of butting the FT up against the CGH, magnifying the FT to overcome a SBWP problem, and using the CT of a square to ensure system alignment and focus. Additionally, linearity of the CGH CT was examined and shown.

Vander Lugt filtering and thermoplastic holography were discussed in Section 3.5. Optical correlation procedures were detailed and examples given. It was noted that an artifact from the CGH CT process could be used in aligning the matched-filter for correlations.

Finally, Section 3.6 briefly discussed digital simulations used to verify optical results. Examples of digitally produced CTs and correlations were provided. Chapter IV follows with results based on experimentation using the optical and digital methods just described.

IV. Experimental Results

4.1 Introduction

This chapter will show and discuss results of the experiments performed to optically implement portions of the Miazza design. In addition, the optical results will be compared to results gained from digitally implementing those same portions of Miazza's design. In the next section of this chapter, results obtained from optical coordinate transforming will be examined. Following this will be a section detailing the results obtained by optically correlating the feature spaces of various input images. Both autocorrelations and crosscorrelations will be analyzed. Finally, a short summary of the results will be given.

4.2 Optical FLRT Feature Space Results

The Fourier transform $\ln r - \theta$ (FLRT) feature space was generated using the techniques detailed in sections 2.4 and 3.4. Figures 32–42 present photographs of several input scenes, their optically generated Fourier transforms (intensity distributions), and their optically generated FLRT feature spaces (coordinate transforms (CTs)), which were generated using the computer generated hologram (CGH). Also included in each figure is the respective computer generated FLRT feature space, which was generated using a digital model of an optical, phase-only filter to perform the coordinate transformation.

Figures 32–34 present the development of the FLRT feature space for a square, a square scaled by one-half, and the scaled square rotated by 45°. Notice that the diffracted orders of the Fourier transform of the smaller square were resolvable in the photograph, compared to the diffracted orders of the transformed larger square which were not resolvable. Actually, resolvability was more a function of the device or medium recording the image. The high contrast, black and white slide film used to record the image intensity distributions (which removed the phase information from the image before further optical processing) demonstrated an ability to resolve the diffracted orders of the Fourier transformed, large square.

In each case investigated, the optical FLRT feature spaces were very similar in appearance to the computer generated FLRT feature spaces. One difference, however, was the optical artifact (ultimately used as a reference point to align FLRT feature spaces for correlation) generated by the CGH that was not present in the digitally generated, optical feature spaces (see Figure 29 in Chapter 3). The absence of this artifact in the digital modeling supports the idea that the artifact is due to the binary-periodic nature of the CGH, which is the binarized interference pattern of the phase-only filter with a plane wave. The binary nature of the CGH allows some of the light to pass through without being diffracted. The light from an image that passes through unhindered is a sampled version of the original image $T(\alpha, \beta)$ incident on the hologram. The sampling can be modeled as the multiplication of an image with a two-dimensional rectangle function (rect) convolved with a two-dimensional array of delta functions commonly called a comb function.

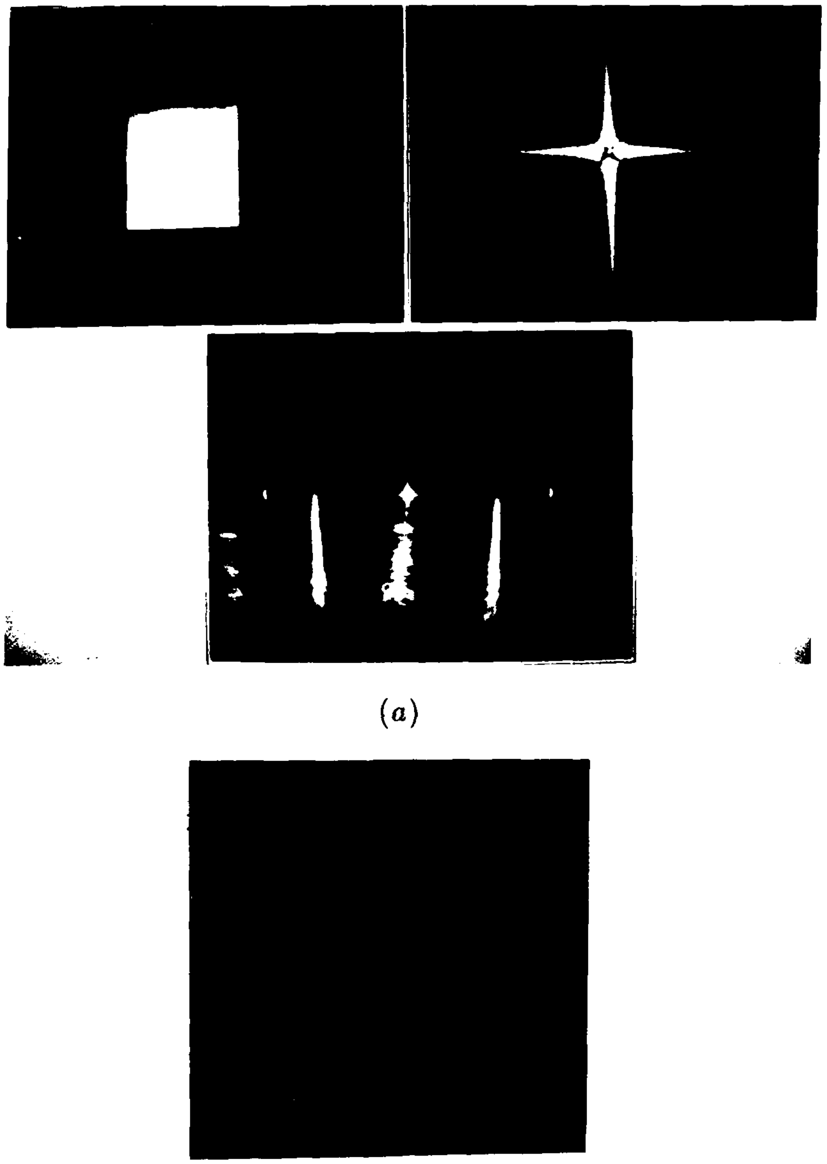
$$T_s(\alpha, \beta) = T(\alpha, \beta)[comb(\alpha, \beta) * rect(\alpha, \beta)] \tag{20}$$

The sampled image is then transformed by the Fourier transforming lens to yield:

$$t_s(f_\alpha, f_\beta) = t(f_\alpha, f_\beta) * [sinc(f_\alpha, f_\beta) comb(f_\alpha, f_\beta)] \tag{21}$$

The final result is the Fourier transform of the image incident on the CGH convolved with a sampled sinc function. Consequently, the resulting pattern is a periodic repetition of the Fourier transformed image modulated by a sinc pattern. For this research, $T(\alpha, \beta)$ was the magnitude squared of the Fourier transform of the original input $t_i(x, y)$. Therefore, the artifact, being the Fourier transform of $T(\alpha, \beta)$, can be mathematically expressed as

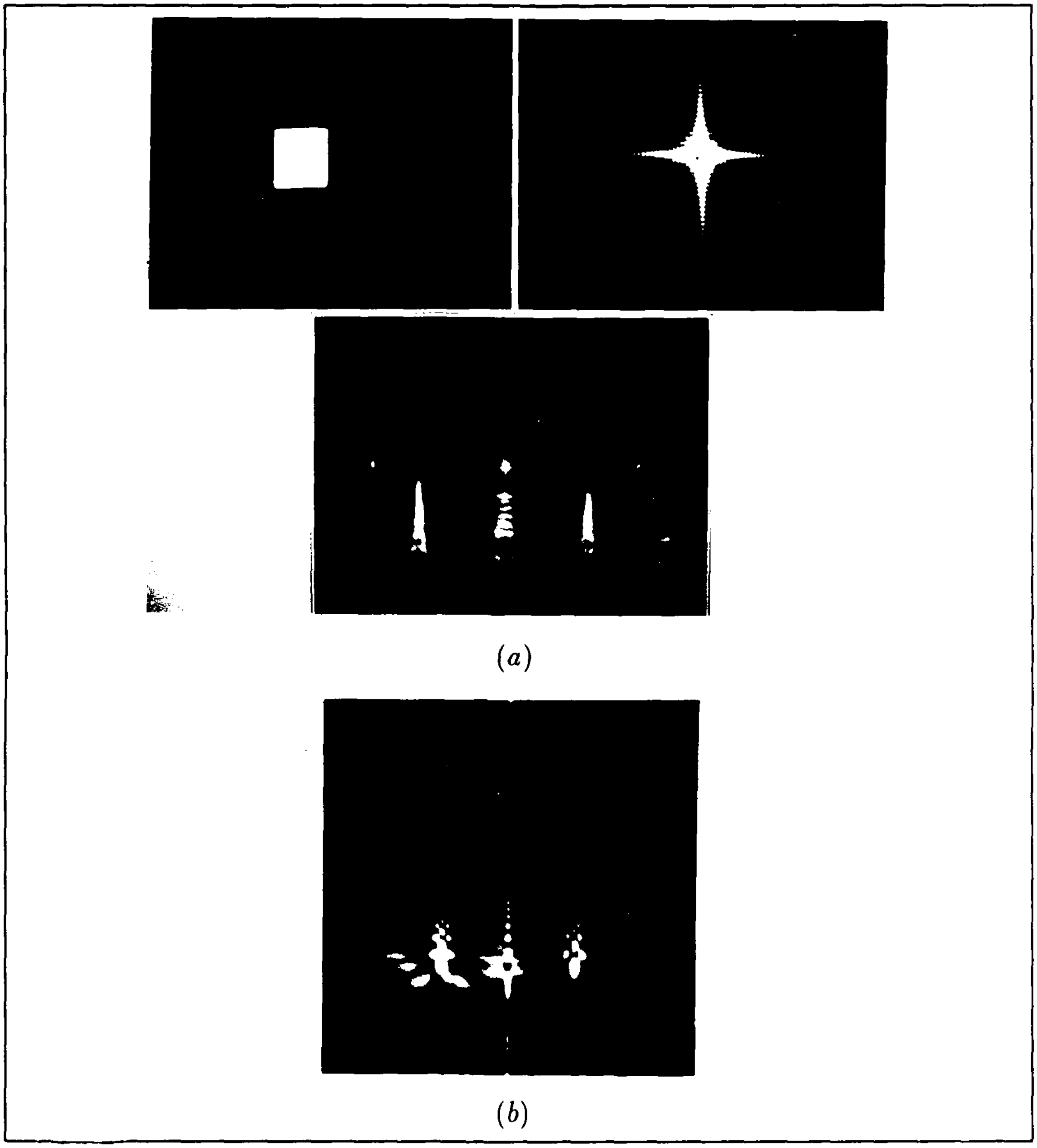
$$\begin{aligned} t(f_\alpha, f_\beta) &= \mathcal{F}[T_i(\alpha, \beta) T_i^*(\alpha, \beta)] \\ &= t_i(x, y) \star t_i^*(x, y) \qquad (22) \end{aligned}$$



(a)

(b)

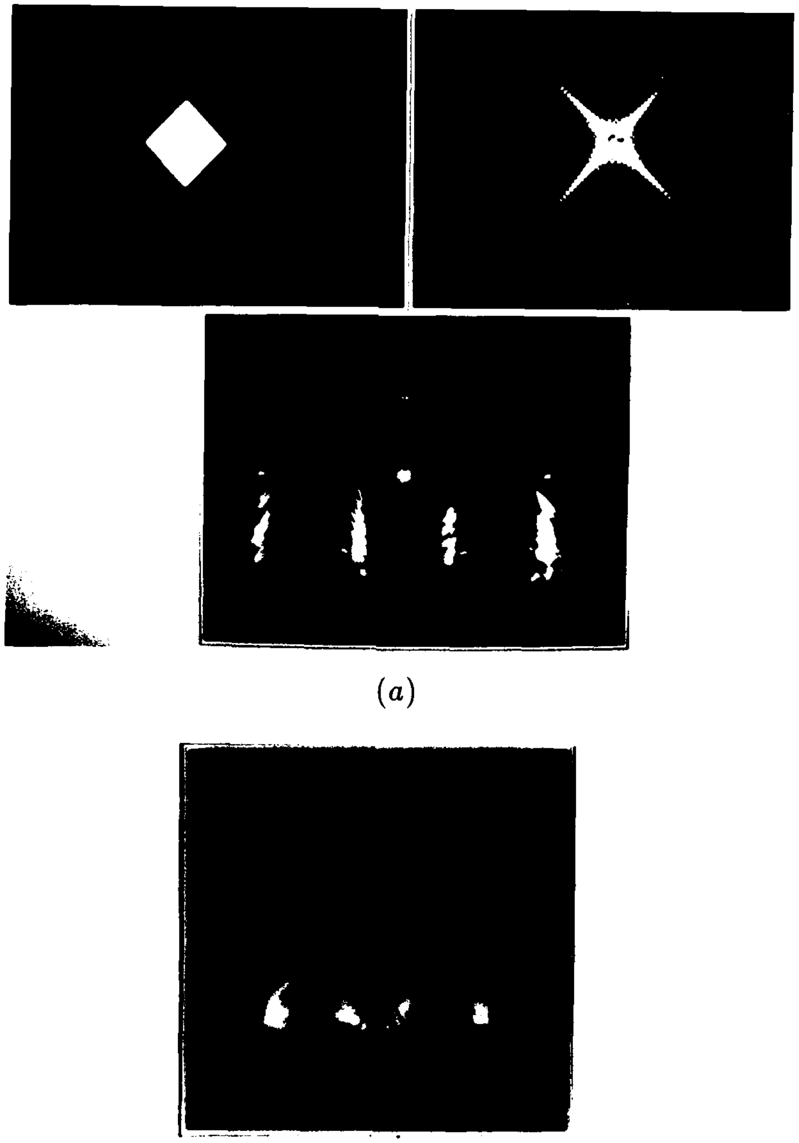
Figure 32. Development of the FLRT CT of a square. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT.



(a)

(b)

Figure 33. Development of the FLRT CT of a square scaled by a half. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT. Note the similarities between Figure 32 and the above FLRT CT even though their exists a scale difference of two between the corresponding FT templates.



(a)

(b)

Figure 34. Development of the FLRT CT of a square scaled by a half and rotated by 45° . Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT. Note: A rotation in the template (see Figure 33) yields a linear shift in the f_θ (horizontal) axis.

which is the autocorrelation of the original input transparency.

A similarity between the optical coordinate transforms and the digitally modeled transforms is an apparent non-linearity in mapping (or unwrapping) of energy along the f_θ (horizontal) axis. In Figure 32(a), the legs of the FT of the square were mapped into three full legs and two half legs along the f_θ axis of the FLRT feature space. The energy of the center leg and the two half legs, which correspond to the uppermost and lowermost legs of the FT, is more dispersed than the two outside (left and right of center) legs. However, in Figure 34, where the square is rotated by 45°, the energy in the FLRT feature space appears to be mapped uniformly across the f_θ axis. These results indicate that energy is not linearly mapped by the CGH along the f_θ axis. The results also indicate that energy incident on the left and right sides of the CGH ($\theta \approx 90^\circ$ or 270°) is mapped tighter than energy incident on the top and bottom portions of the CGH. Furthermore, energy incident at the four corners of the CGH ($\theta \approx 45^\circ$, 135°, 225° and 315°) appears to be evenly mapped. Note, however, that the legs of the optical CT shown in Figure 34 have beveled tops. This result occurs because the FT had to be magnified to overcome the SBWP problem discussed in section 3.4.4. When the Fourier transform was magnified, the spatial extent of its legs was such that some of the higher diffracted orders were cut off by the edges of the CGH. Therefore, that lost energy was not coordinate transformed. In the particular case of the rotated square, the four legs of the FT were each cutoff at a 45° angle, and thus, the legs of the CT are beveled at 45°.

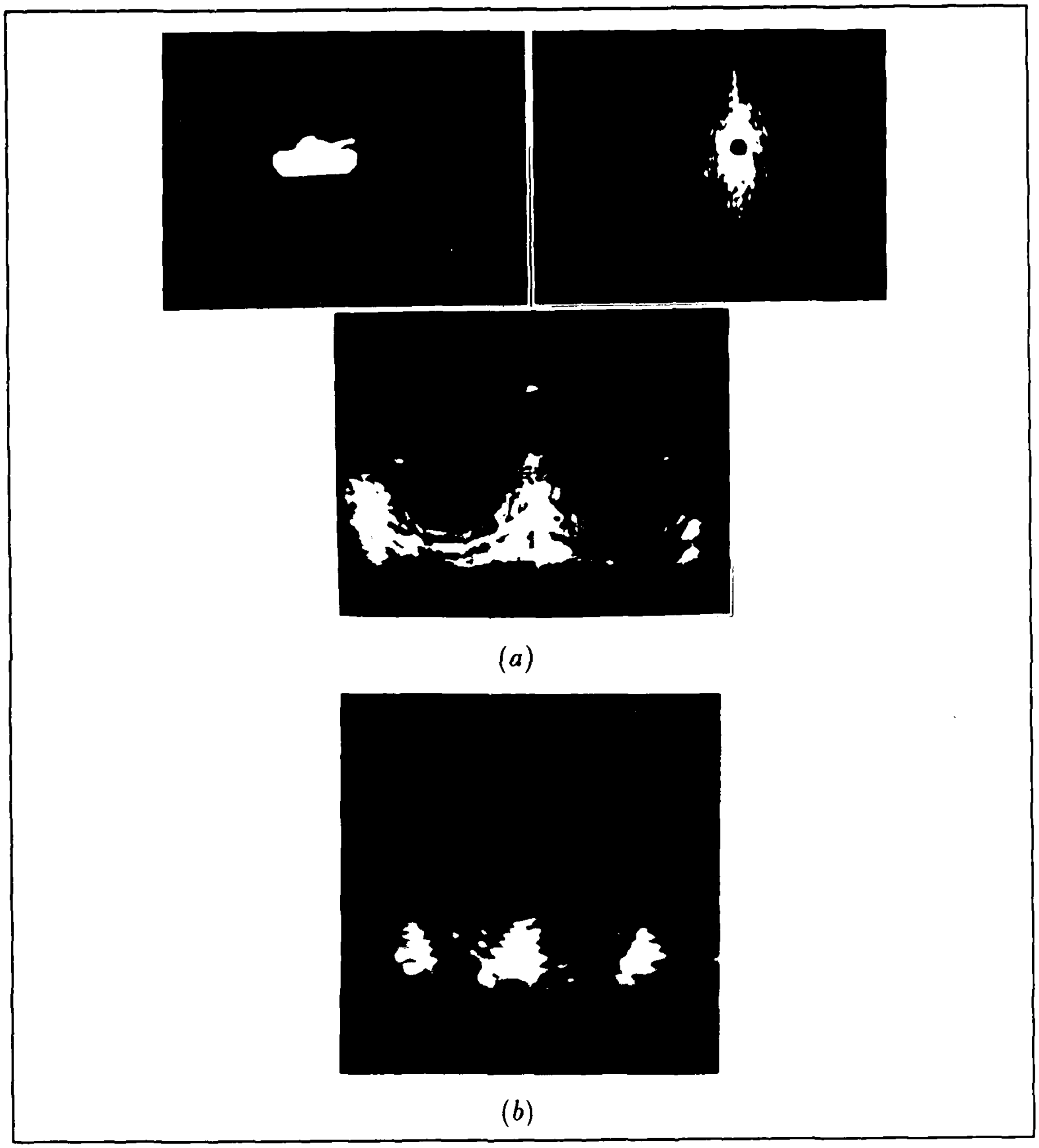
Note also, that a rotation in the input generates a cyclic shift along the f_θ axis of the FLRT feature space. In Figure 32, the distance between the legs of the FT along the f_θ axis of the FLRT feature space is 90°. The FLRT feature space in Figure 34 shows that a rotation in the input by 45° causes a cyclic shift of the feature space by a distance equivalent to half the distance between adjacent legs in the feature space, or 45°. Therefore, the half legs observed in Figure 32 cyclicly shift and combine to yield four whole legs in the rotated square's feature space.

Figures 35–42 are FLRT feature spaces of military vehicles and aircraft. Figures 35 and 36 present the development of the FLRT feature spaces of a tank and a tank scaled by three-fourths. The input template of the tank is very similar in geometry to a rectangle and an ellipse. Consequently, the intensity distribution of the *FT* of a tank can be analyzed based on similarities to these simple geometric shapes. As shown in Figure 35, the *FT* of a tank produces a prominent vertical narrow band of energy caused by its narrow height compared to its length. Also, there is a small amount of energy due to the tank's narrow barrel being transformed at an angle to the vertical band, but it is difficult to discern from the photograph. The tank's elliptical shape generates *FT* energy in a scattered elliptical, quasi-Airy disk pattern. Consequently, the FLRT feature space of a tank has one full leg in the middle corresponding to the energy at the top of the *FT* and two partial legs, which when combined, correspond to the vertical band of energy at the bottom of the *FT*. Since all of the vehicle templates tested had a general rectangular shape, their *FT*s and corresponding FLRT feature spaces were all similar in appearance, with each feature space having two prominent legs of energy. Figures 39–42 present the FLRT feature spaces of input scenes with multiple image types present.

4.3 Correlation Results

The main thrust of this research was to show and verify the predictability of optically correlating a position, scale, and rotation invariant feature space using Vander Lugt matched-filters. The predictable characteristic of the correlation was peak location. A specific horizontal shift would indicate a corresponding change in the input image's relative rotation, and a specific vertical shift would indicate a corresponding change in the input image's relative scale.

Figures 43–48 present examples of optical correlations of optically generated FLRT feature spaces and digital correlations of computer-modeled, optical FLRT feature spaces (presented in the previous section), as well as, three-dimensional plots



(a)

(b)

Figure 35. Development of the FLRT CT of a tank. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT.

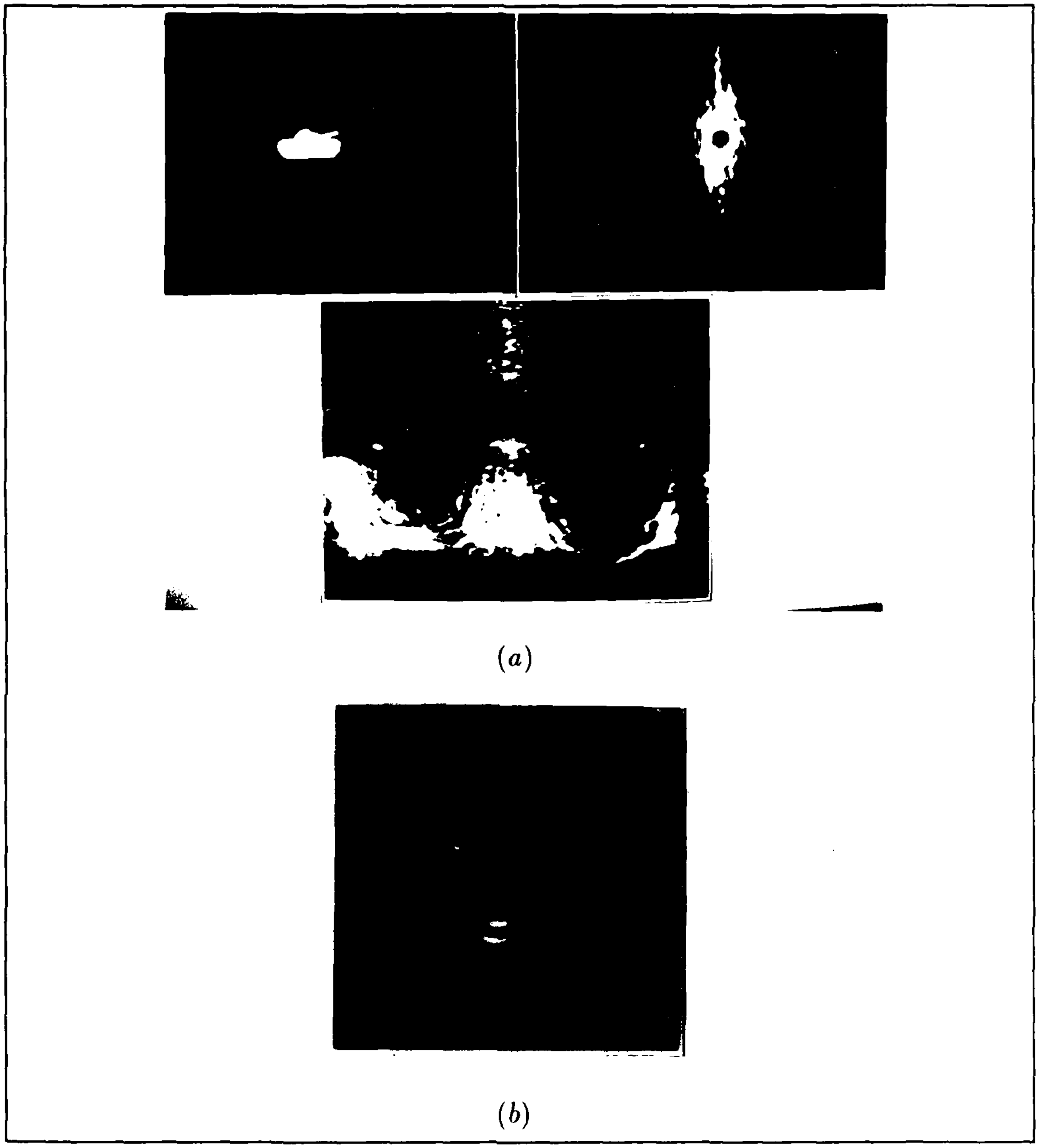
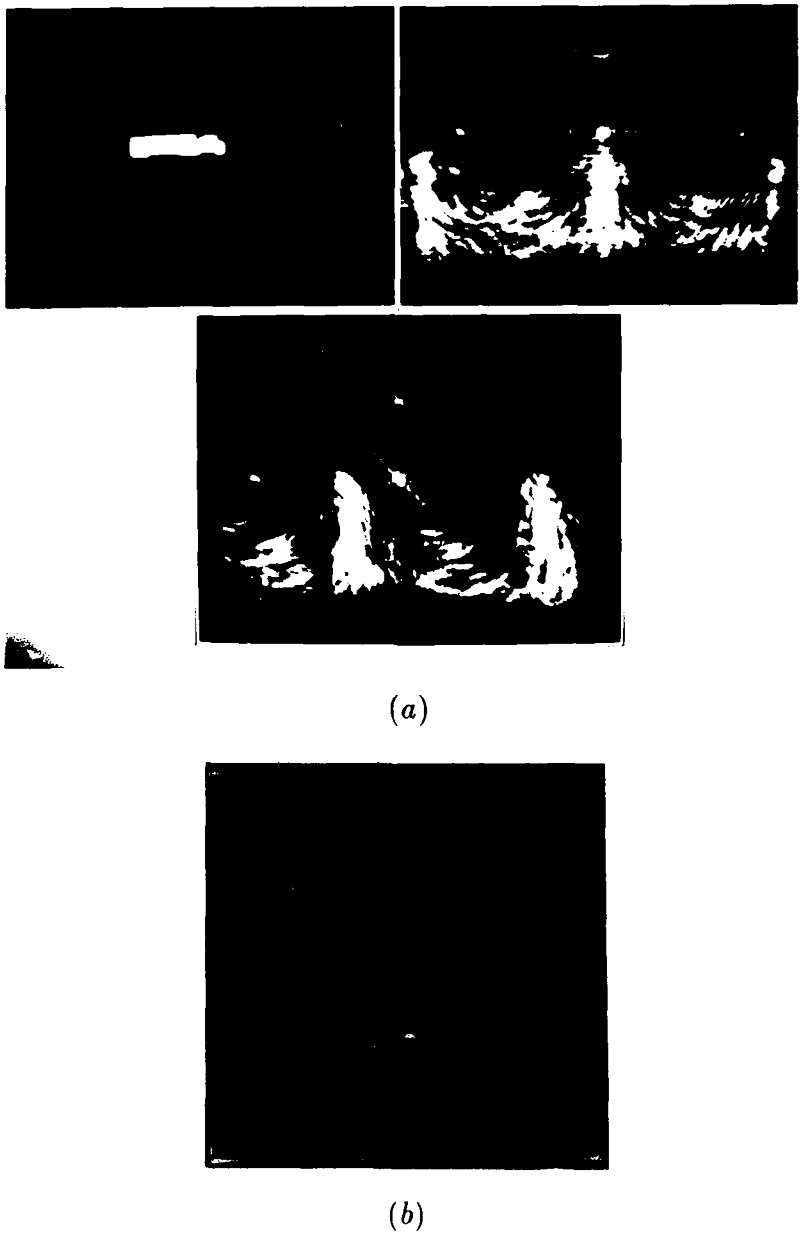


Figure 36. Development of the FLRT CT of a tank scaled by a half. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT. Note the similarities between Figure 35 and the above FLRT CT even though there exists a 3/4 scale difference between the corresponding FT templates.



(a)

(b)

Figure 37. FLRT CT of a truck. Photograph (a) is the input template (top left), the optically generated FLRT CT template (top right), optically generated FLRT CT of truck rotated by 45° (bottom). Photograph (b) is the digitally modeled optical FLRT CT. Note: Multiple object scenes were tested for target discrimination performance.

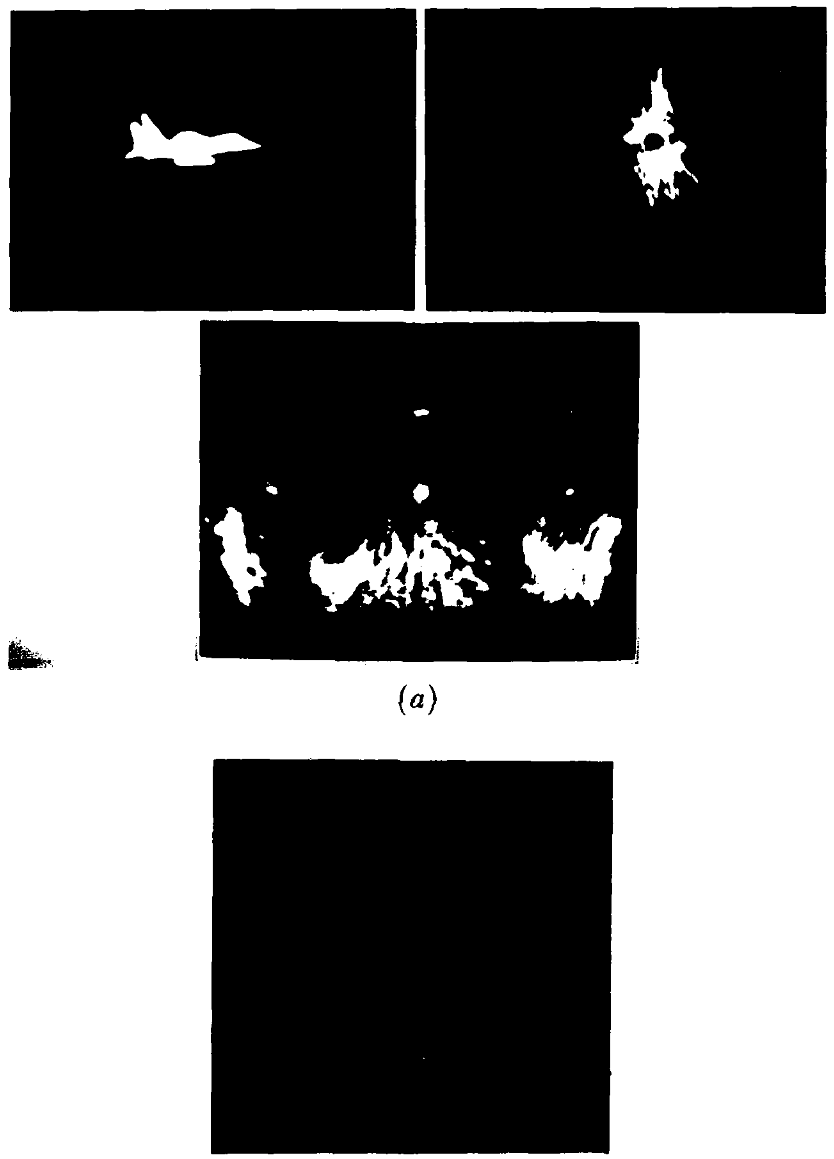
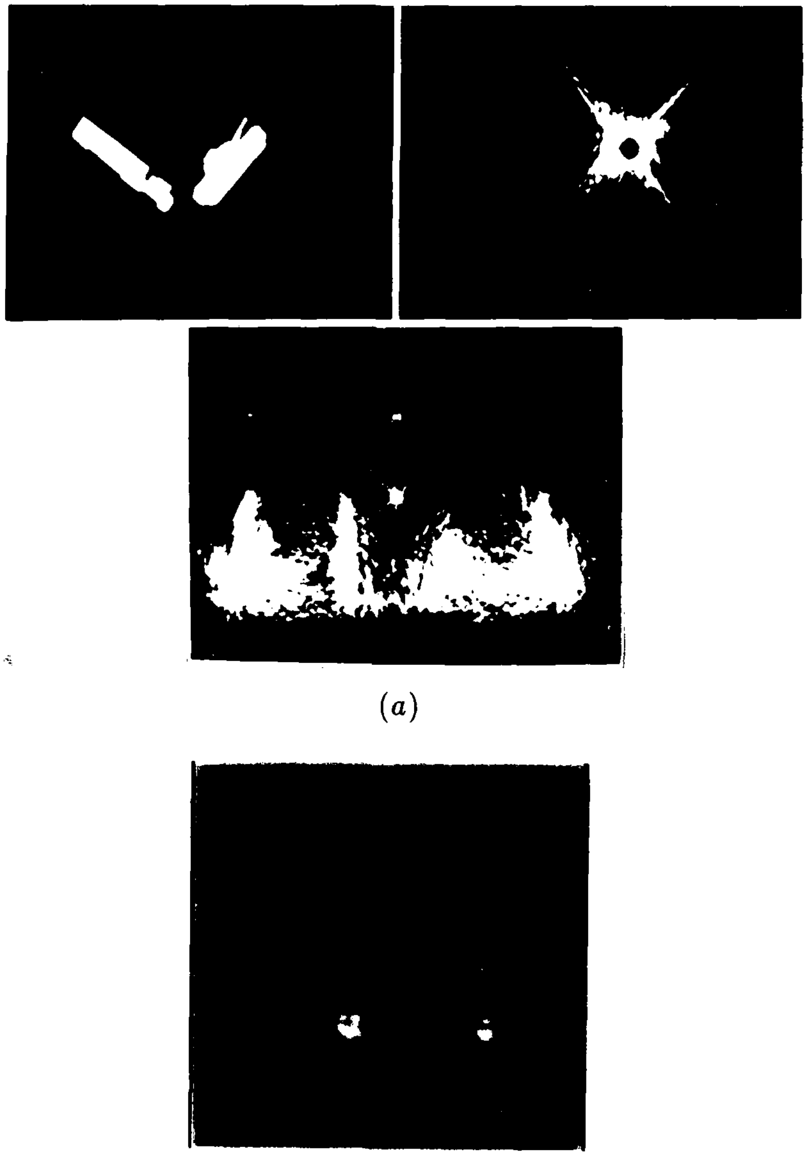


Figure 38. Development of the FLRT CT of an F-15. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT.



(a)

(b)

Figure 39. Development of the FLRT CT of a truck and tank rotated 90° with respect to each other. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT.

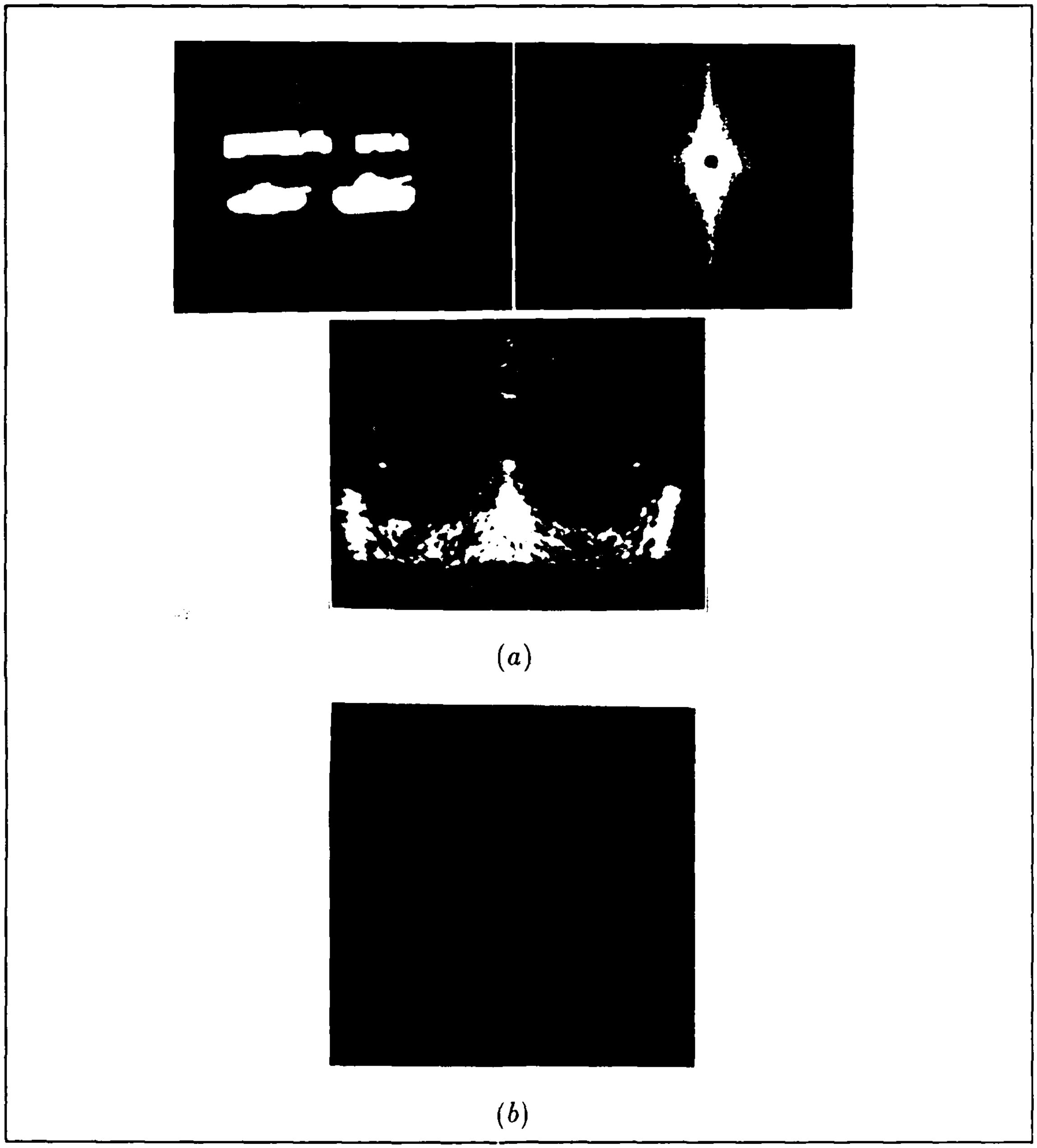
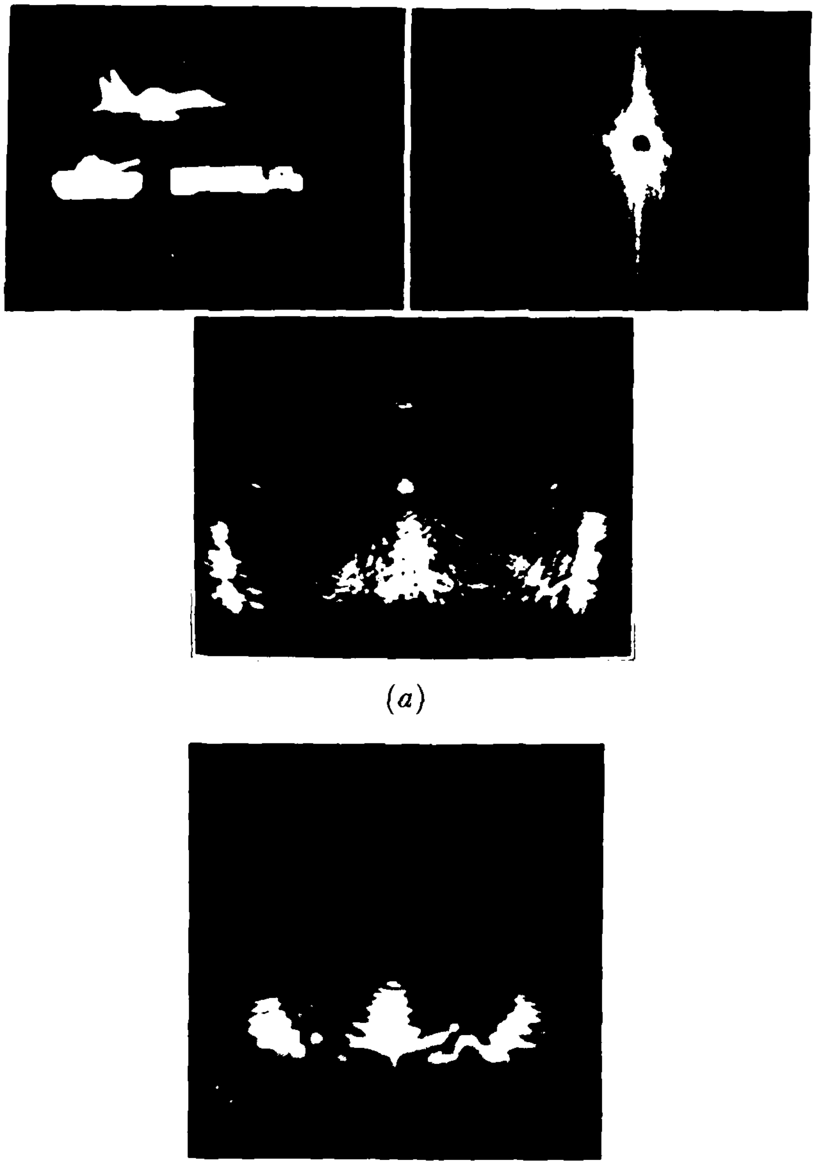


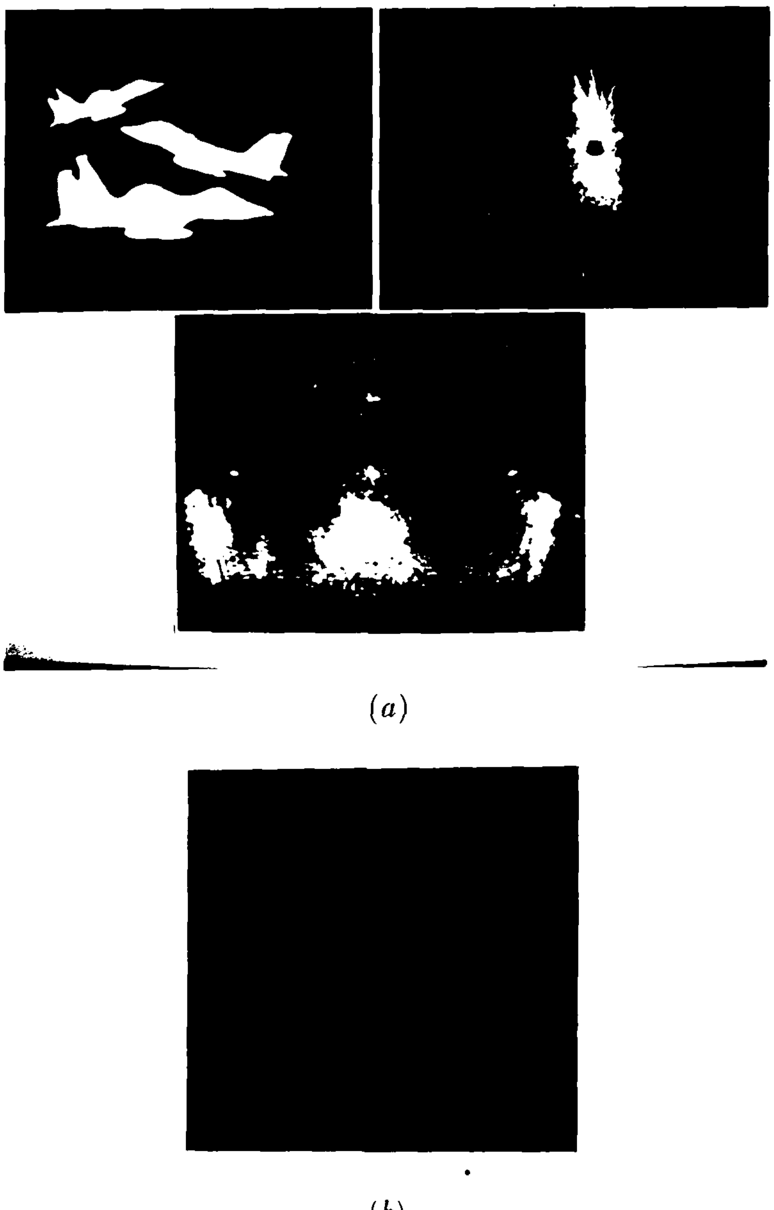
Figure 40. Development of the FLRT CT of two tanks and two trucks. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT.



(*a*)

(*b*)

Figure 41. Development of the FLRT CT of a tank, truck, and F-15. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT.


(a)

(b)

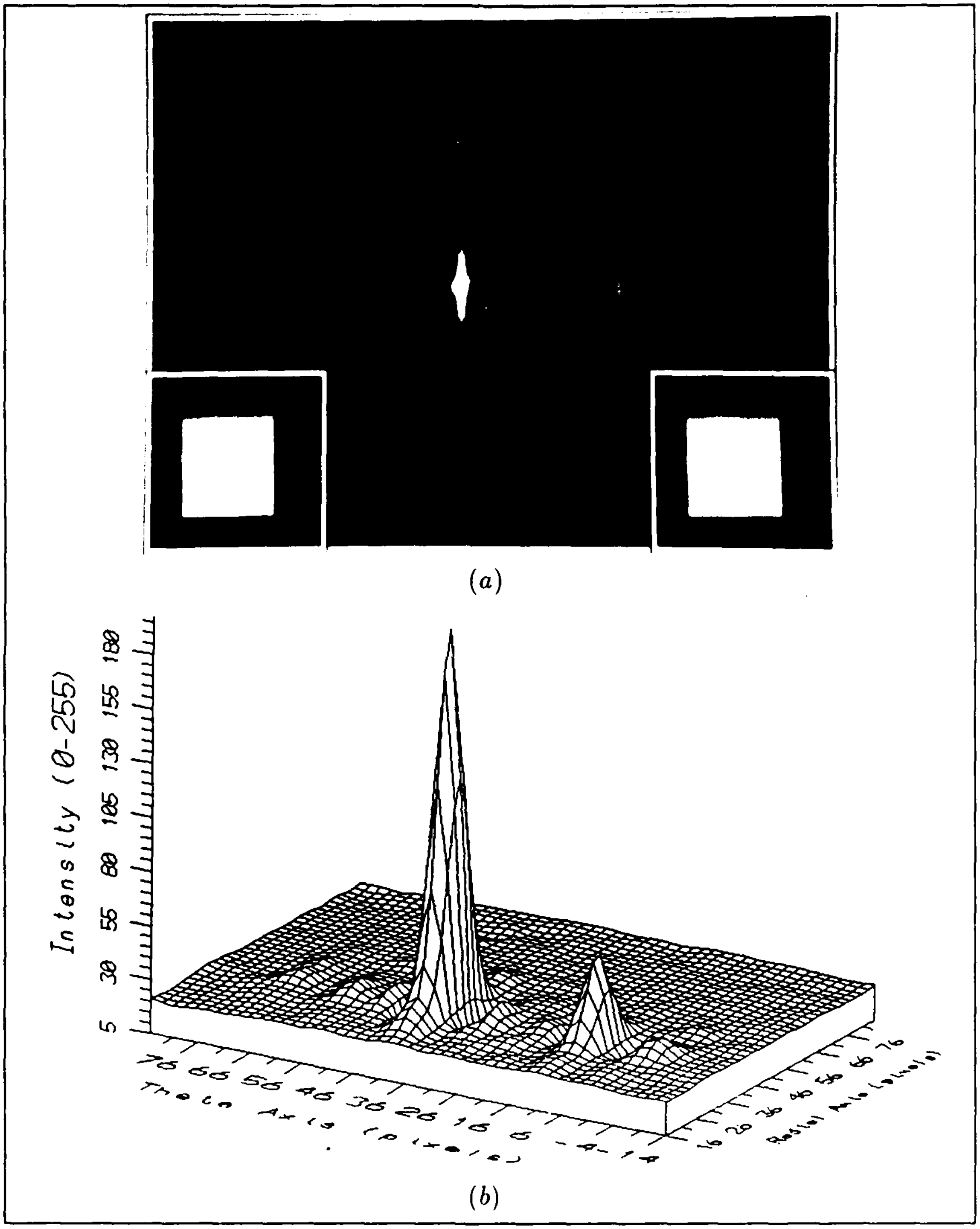
Figure 42. Development of the FLRT CT of three F-15's. Photograph (a) is the input template (top left), the optical FT of the template (top right), optically generated FLRT CT (bottom). Photograph (b) is the digitally modeled optical FLRT CT.

of each correlation. The optical correlations were performed using the Vander Lugt Filtering technique outlined in section 3.5. The computer-modeled, optical FLRT feature spaces were correlated using a linear correlation routine written by Kobel and Martin. Displayed in the lower-left and lower-right corners of the correlation intensity distribution photographs are the templates used to generate the feature spaces that produced the correlation. On the left is the template used to generate the Vander Lugt filter, and on the right is the input template.

Optical correlations were attenuated using neutral density filters to avoid saturating the CCD camera. The attenuation was held constant throughout the experiment so the relative intensities of cross-correlation and autocorrelation peaks could be compared. However, the digital correlation data was normalized independently for each correlation and compressed into 256 discrete levels. Consequently, the digital data could be evaluated only on the merits of peak shift and shape, and not on relative height (intensity).

The 3-dimensional intensity plots of the correlations were created by digitally reducing the correlation images down to 128 by 128 pixel images. These smaller images were then plotted using a laser printer and a plotting routine named SURFER.

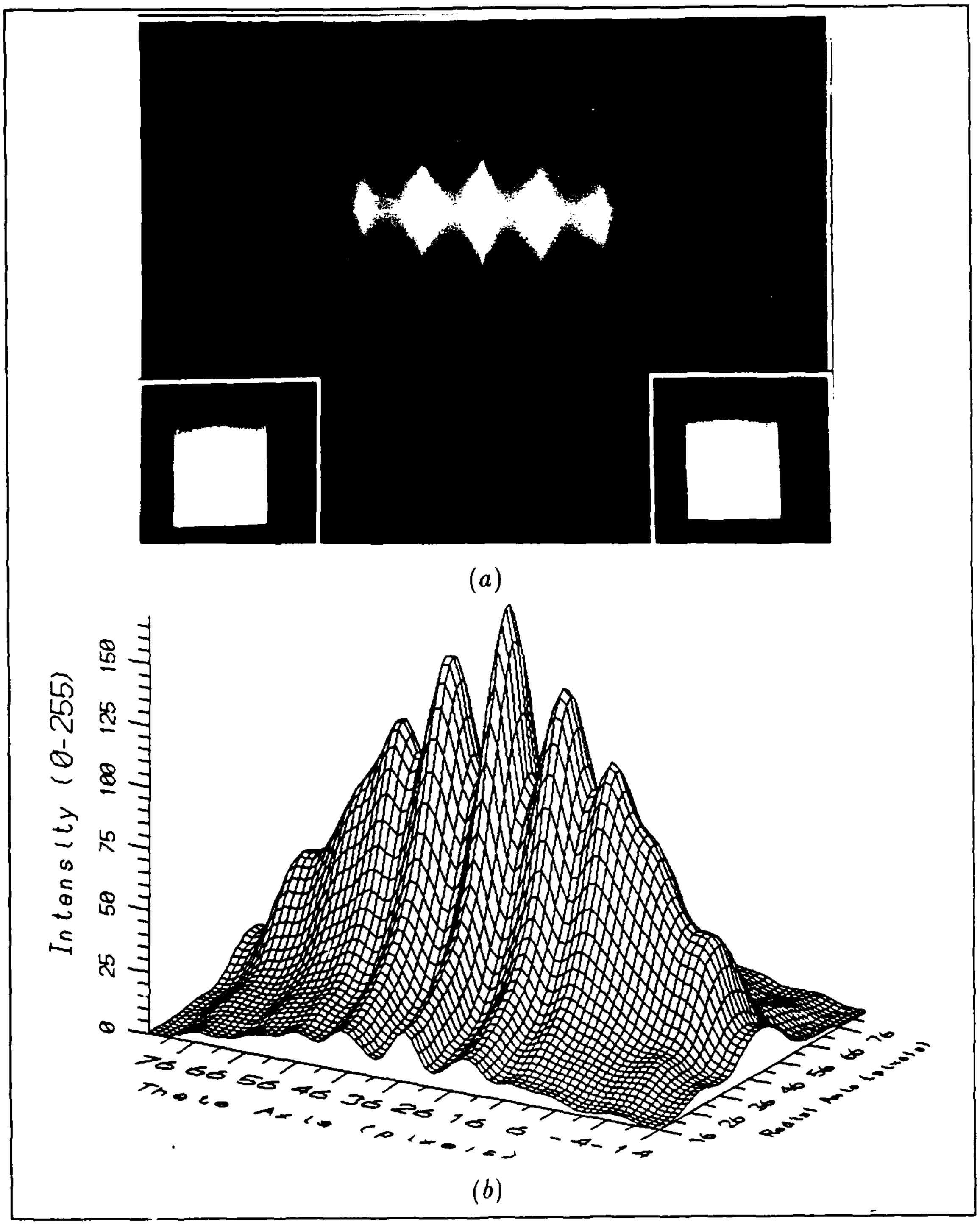
4.3.1 Target Rotations Figures 43–45 demonstrate the classic linear translation of a correlation peak along the f_θ axis due to an in-plane rotation of a square by 45° . Relative rotational shifts were determined by measuring the the distance the correlation peak translated along the f_θ axis with respect to the autocorrelation peak obtained by correlating the template CT used to make the matched-filter with the matched-filter. The f_θ axis shift was calibrated based on the relative distances between peaks where the relative angles are known. Since the Fourier transform of a square has "legs" at equal increments of 90° apart, both the FLRT feature space and the autocorrelation plane have periodic peaks 90° apart. This particular



(*a*)

(*b*)

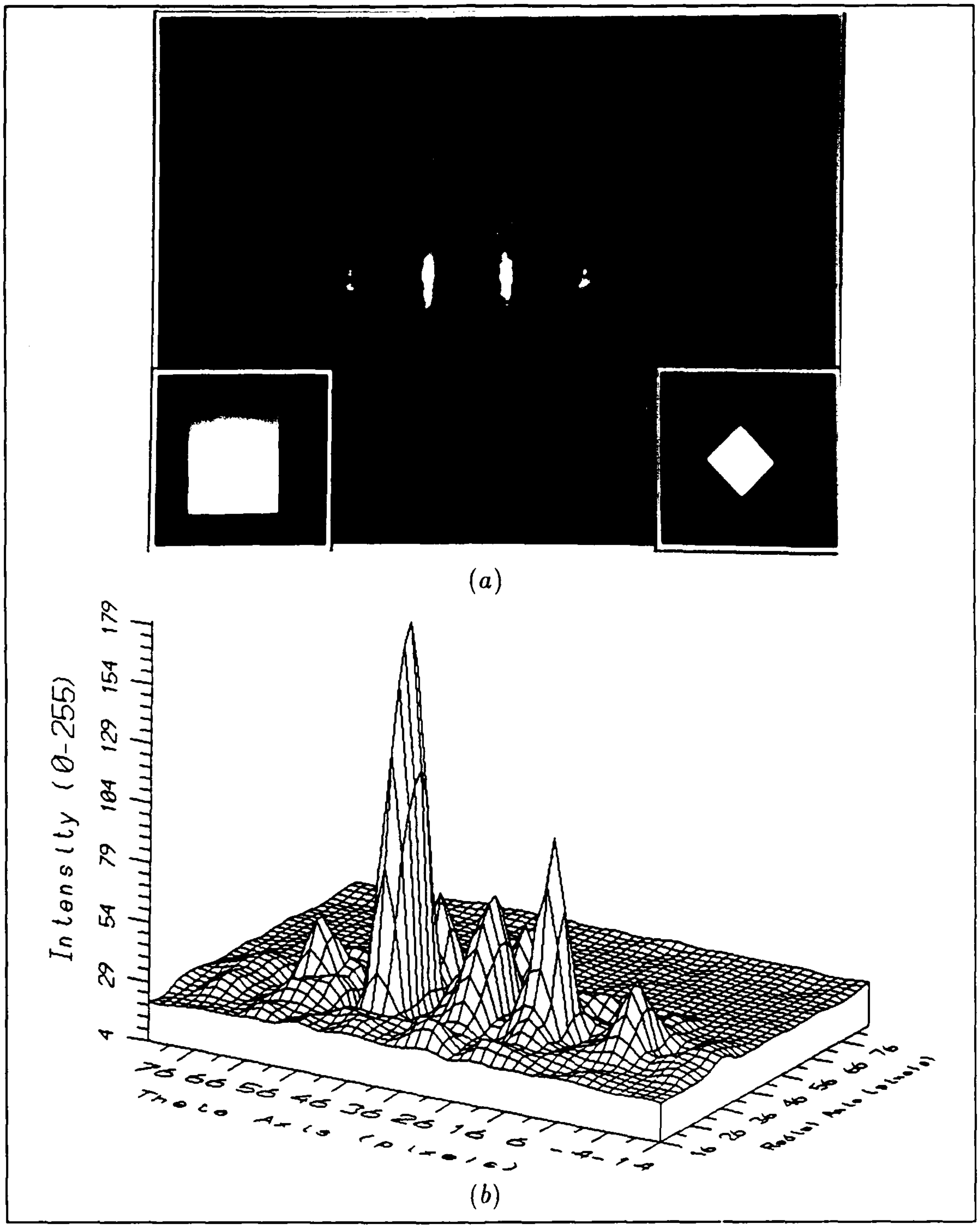
Figure 43. Optical autocorrelation of the FLRT CT of a square where (a) is a photograph of the intensity distribution, and (b) is a 3-D plot. Note: Correlations were attenuated to avoid saturating CCD camera.



(a)

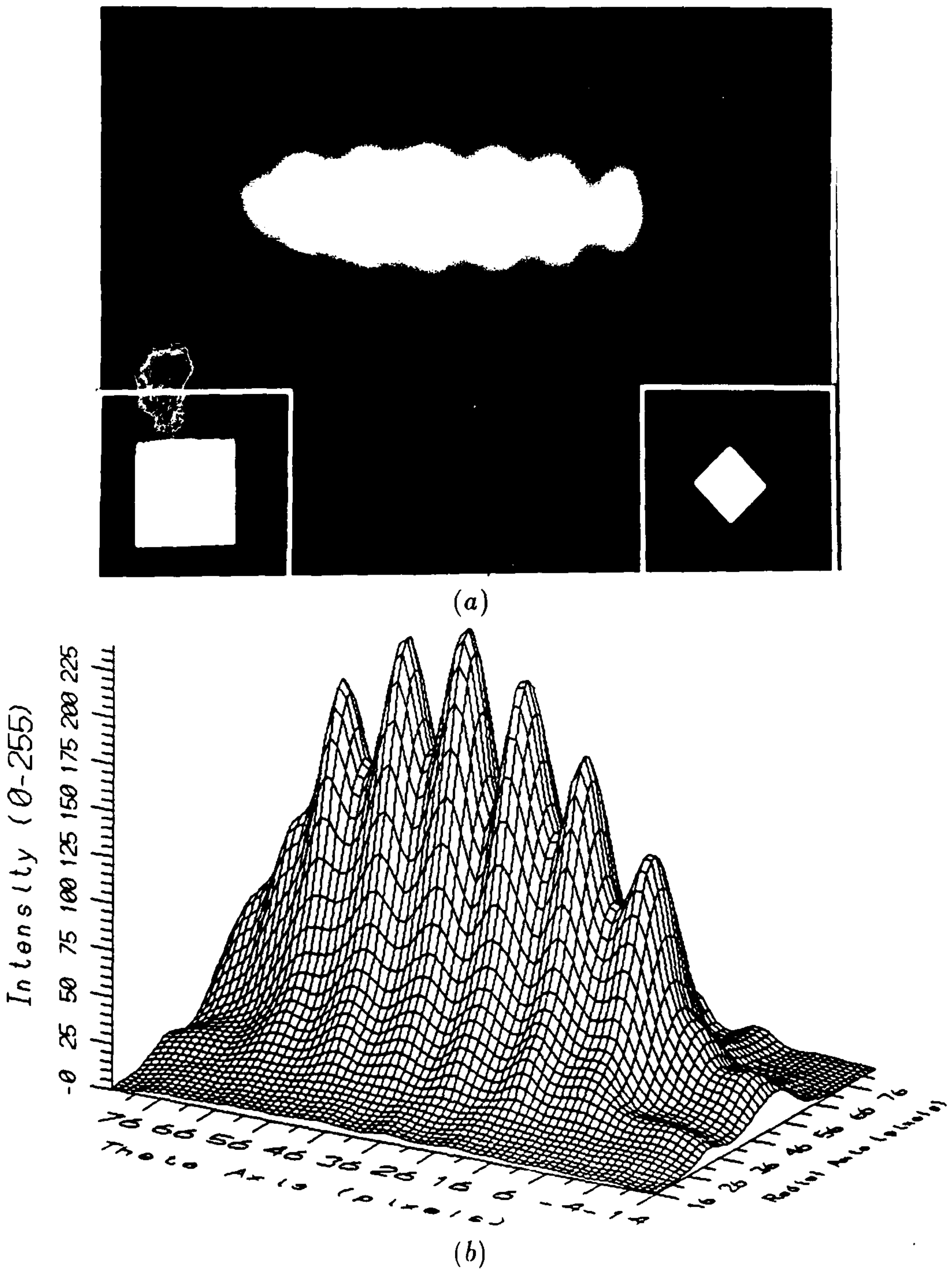
(b)

Figure 44. Digital autocorrelation of the computer-modeled optical FLRT CT of a square where (a) is a photograph of the intensity distribution, and (b) is a 3-D plot. Note: Digital correlation data was normalized to 256 intensity levels.


(a)

(b)

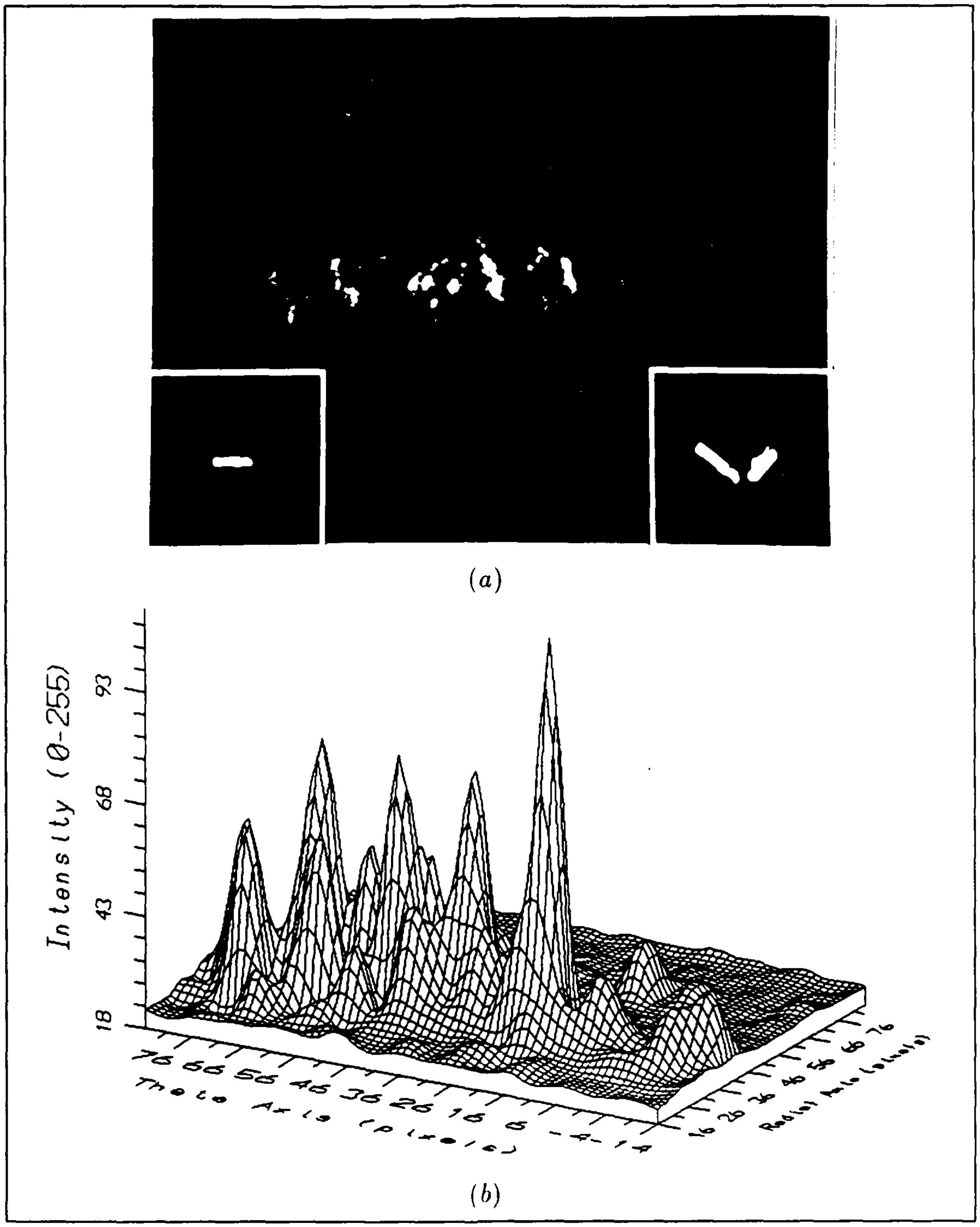
Figure 45. Optical correlation of the FLRT CTs of a square and a square scaled by a half and rotated 45° where (a) is a photograph of the intensity distribution, and (b) is a 3-D plot. Note correlation peak shifted 46.1° compared to the autocorrelation peak in Figure 43.



(a)

(b)

Figure 46. Digital correlation of the computer-modeled optical FLRT CTs of a square and a square scaled by a half and rotated 45° where (a) is a photograph of the intensity distribution, and (b) is a 3-D plot. Note correlation peak shifted 46.9° compared to 46.1° optically (see Figure 45).



(a)

(b)

Figure 47. Optical correlation of the FLRT CTs of a truck and of a truck and tank rotated 45° in opposite directions where (a) is a photograph of the intensity distribution, and (b) is a 3-D plot. Note: Due to cyclic nature of θ axis, the pseudo-autocorrelation peak is located at $-42.7°$ versus the highest peak located at $129.5°$.

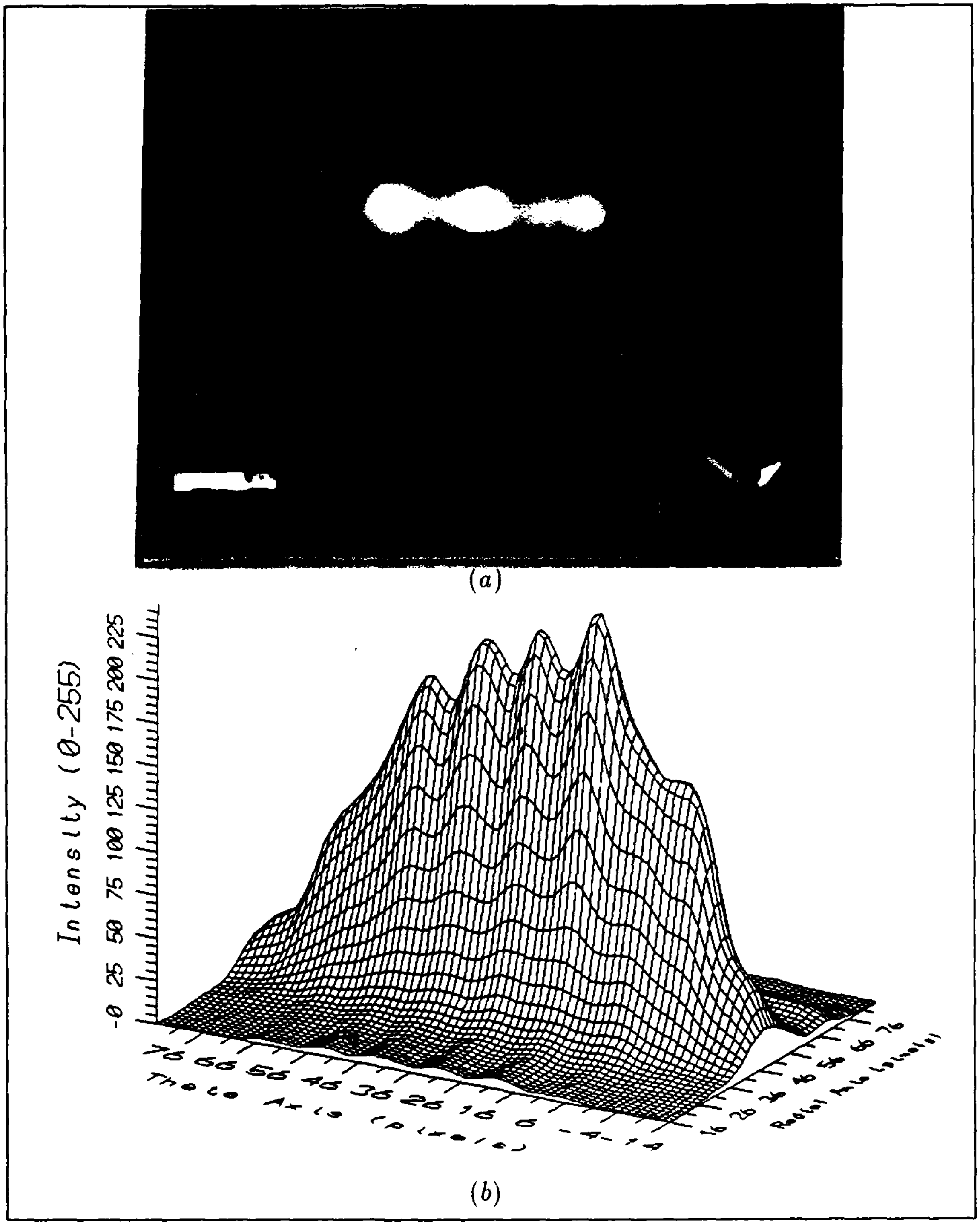


Figure 48. Digital correlation of the computer-modeled optical FLRT CTs of a truck and a truck and tank rotated 45° in opposite directions where (a) is a photograph of the intensity distribution, and (b) is a 3-D plot. Similar to Figure 47, the pseudo-autocorrelation peak (second highest) is located at −51.6° versus highest peak at 137°.

autocorrelation (see Figure 43) was used to calibrate the f_θ axis. The correlation between the CT of a square rotated 45^o and the matched-filter of an unrotated square yielded a relative rotational shift of 46.1^o when performed optically and 46.9^o when performed digitally.

Figures 47 and 48 reveal a weakness in optically correlating the FLRT feature space. These figures show the correlation plane of a truck CT correlated with the CT of a truck and a tank rotated 45^o in opposite directions. Similar to the case of the rotated square presented above, one might expect the rotated truck correlation plane to have its largest peak located -45^o away from the truck's autocorrelation peak. With calculations based on the largest peak, the correlation plane indicates a shift of 129.5^o when generated optically and 137.0^o when generated digitally. The peak that actually corresponded to the -45^o shift was the third highest in intensity and indicated a rotation of -42.7^o when performed optically and -51.6^o when performed digitally.

These false indications exist because rotations in the input space yield cyclic shifts in FLRT space, whereas optical correlations are linear. Consequently, correlations performed in the FLRT space must be cyclic along the f_θ axis to ensure that the feature spaces can completely match to yield a true autocorrelation peak. Figure 49 graphically illustrates the problem of linearly correlating a cyclic feature space. Part (a) illustrates a generic CT as seen in the current 2π space, while (b) shows how the CT of part (a) would look in a 4π space. Then, in part (c) the result of autocorrelating the CT of (a) is shown. It is this true autocorrelation peak (as shown by the arrow in (c)) that is the desired result when correlating position, scale, and rotation invariant feature spaces. With respect to rotation, part (d) shows how the generic CT would change when the input image is rotated by 45^o. Now, when the rotated version, (d), is correlated with the matched-filter, (a), the resultant peak is no longer a true autocorrelation. This is illustrated in part (e) of Figure 49, where linear correlation of a cyclic feature space is shown to produce an asymmetri-

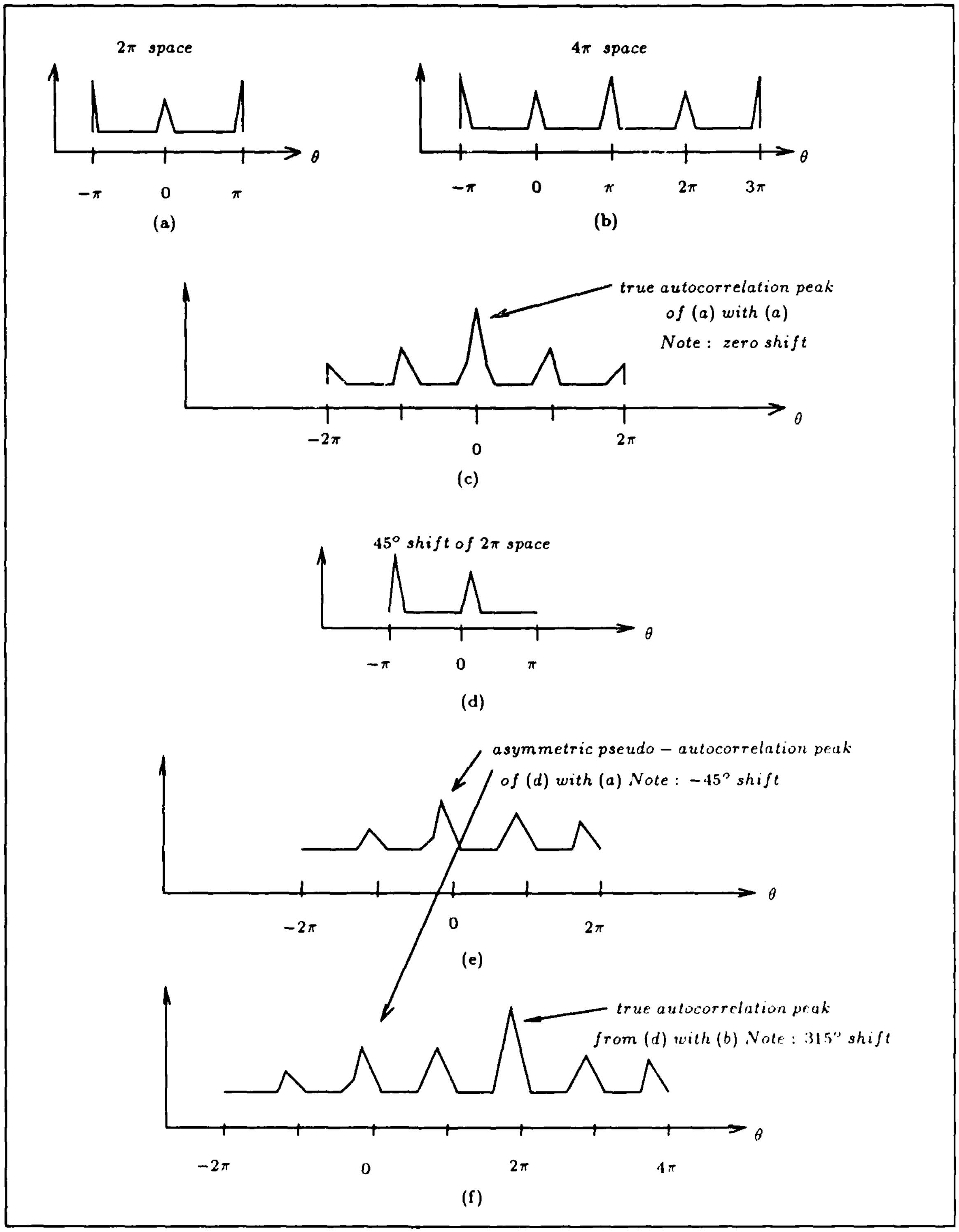


Figure 49. Graphic illustration of 4π versus 2π optical correlation where (a) is an example of 2π space and (b) is an expanded 4π space of (a). Graph (c) is autocorrelation of (a). Graph (d) is (a) shifted clockwise 45° . Graph (e) is correlation of (a) with (d). Graph (f) is correlation of (d) with 4π space (b).

cal, pseudo-autocorrelation peak. Asymmetry occurs since the feature space wraps around when the input image is rotated, resulting in a condition where the rotated feature space may not perfectly overlap the matched-filter during the linear correlation. Casasent proposed that this phenomena could be eliminated by correlating the 2π input feature space with an expanded 4π matched-filter feature space [1:940]. Figure 49(f) illustrates that correlating with an expanded feature space produces a symmetrical, autocorrelation peak.

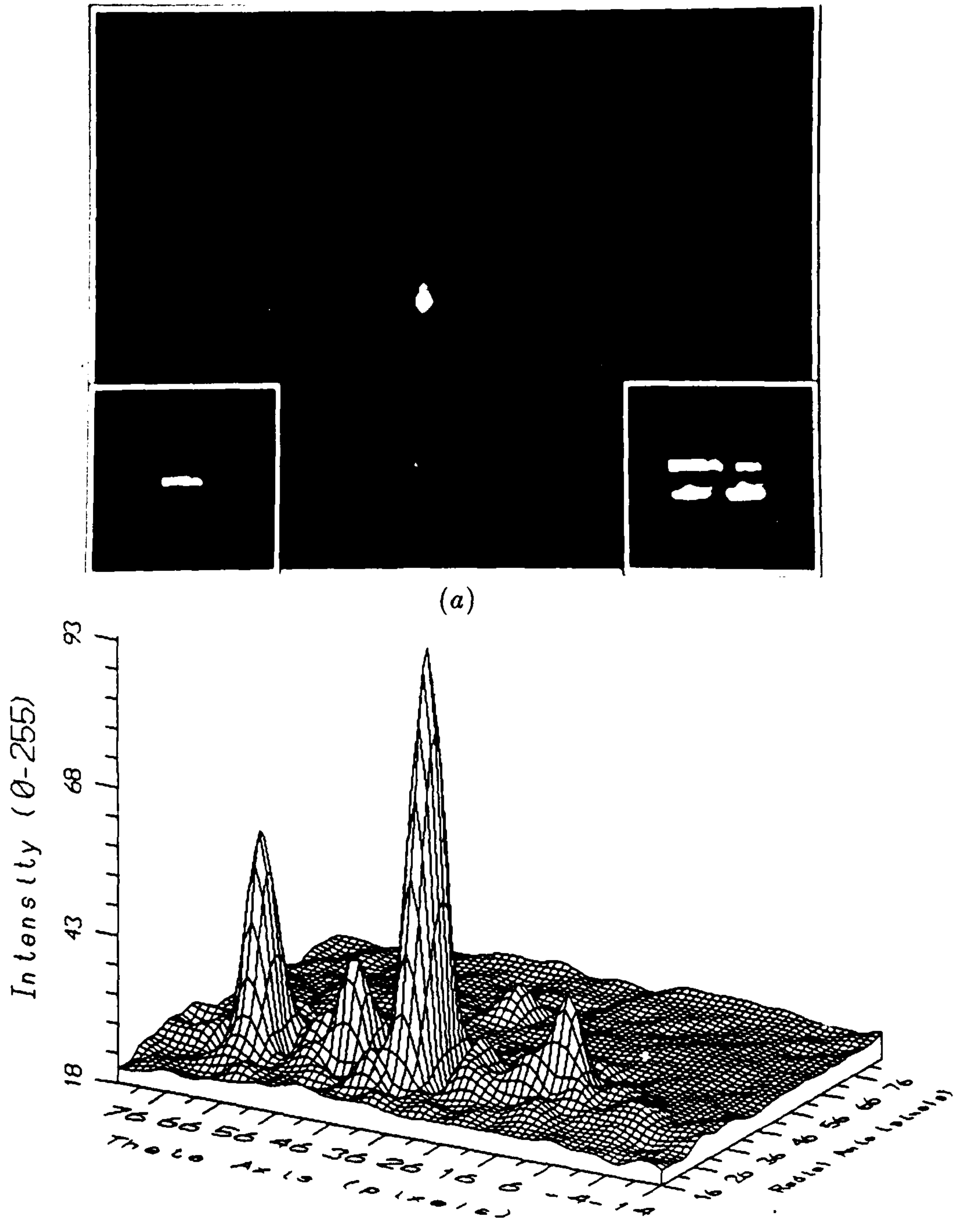
4.3.2 Target Scale Measurable translations of correlation peaks along the radial axis were observed only for a square scaled by half (see Figures 43 and 45). In this case, the correlation peak shifted 20 pixels (in a vertical field of 400 pixels) along the radial axis. According to data presented in section 3.4.5 on the logarithmic response of the CGH (see Figure 21), a scale change of two on the nonlinear portion of the graph will shift the output feature space by approximately 23mm in an field of 250mm *(the size of the display used in this research)*. On the 400 pixel vertical field of the CCD camera, 23mm would correspond to a shift of 36.8 pixels. However, in the optical correlation setup used in this study (see Figure 27), the sizes of the input and output Fourier transforming lenses (lenses 1 and 2) were chosen to scale down the output correlation by a factor of two. Therefore, the theoretical shift of the correlation peak was 18.4 pixels, which compares quite favorably to the 20-pixel shift observed.

But why were correlation peak translations due to scale changes only observed for templates of squares? Due to the nonlinear response of the CGH on the radial axis, the scale change of a small input image incident on the CGH exhibits a greater shift in the feature space than an equivalent scale change of a much larger image. The CGH exhibited a greater shift in the feature space when operating in the nonlinear portion of the logarithmic curve shown in Figure 21 (b). The square templates used during the research were much larger than the tank, truck, and aircraft templates. Consequently, the Fourier transform spaces of the squares were notably smaller than the transform spaces of the other templates. The larger Fourier transform spaces were coordinate transformed on the linear portion of the CGH's logarithmic curve, where changes in scale result in very small translations of the output.

Ability to measure scale translations in the FLRT feature space is also dependent on the resolution of the of the detector used in the correlation plane. For this research, a CCD camera with a digital framegrabber was used as the correlation plane detector. The CCD camera has a vertical range of 400 pixels. In the case of the scaled square example, the peak only translated 5 percent of the vertical range of the camera. A cylindrical lens could be used to expand the correlation output in the radial axis to increase the detector's ability to resolve the smaller shifts along the radial axis.

4.3.3 Target Detection While the previous sections discussed determining scale and rotation changes of input images based on movement of the autocorrelation peak, this presupposes that the FLRT feature space will provide correlation planes in which auto- and cross-correlations can be distinguished from one another. In other words, if the input scene contains multiple images (or other sources of background noise), will the FLRT space correlations provide adequate resolution to allow determination of target presence? Figures 50–55 present results of this research that provide some insight into answering the above question. In Figures 50–52, the photographs in (a) and the plots in (b) represent correlation planes obtained optically where the input scene contained the target template, as well as, additional images.

Ideally, one would expect the most intense peak to be a symmetric, autocorrelation peak positioned appropriately based on any scale or rotation changes. While, Figures 50 and 51 do show the highest intensity peaks at the expected, zero-shift location, Figure 52 has its highest peak located approximately 180° from the expected location. It should be noted that the apparent lack of crosscorrelation components in Figure 51 is due to an increased amount of neutral density filtering used when



(a)

(b)

Figure 50. Optical correlation of the CTs of a truck and multiple vehicles where (a) is the intensity distribution and (b) is the 3-D plot. Note: No shift in θ axis indicates 0^o rotation. Peak shifted 5 pixels in radial axis compared to theoretical 6.8 pixels (based on linear portion of CGH ln curve).

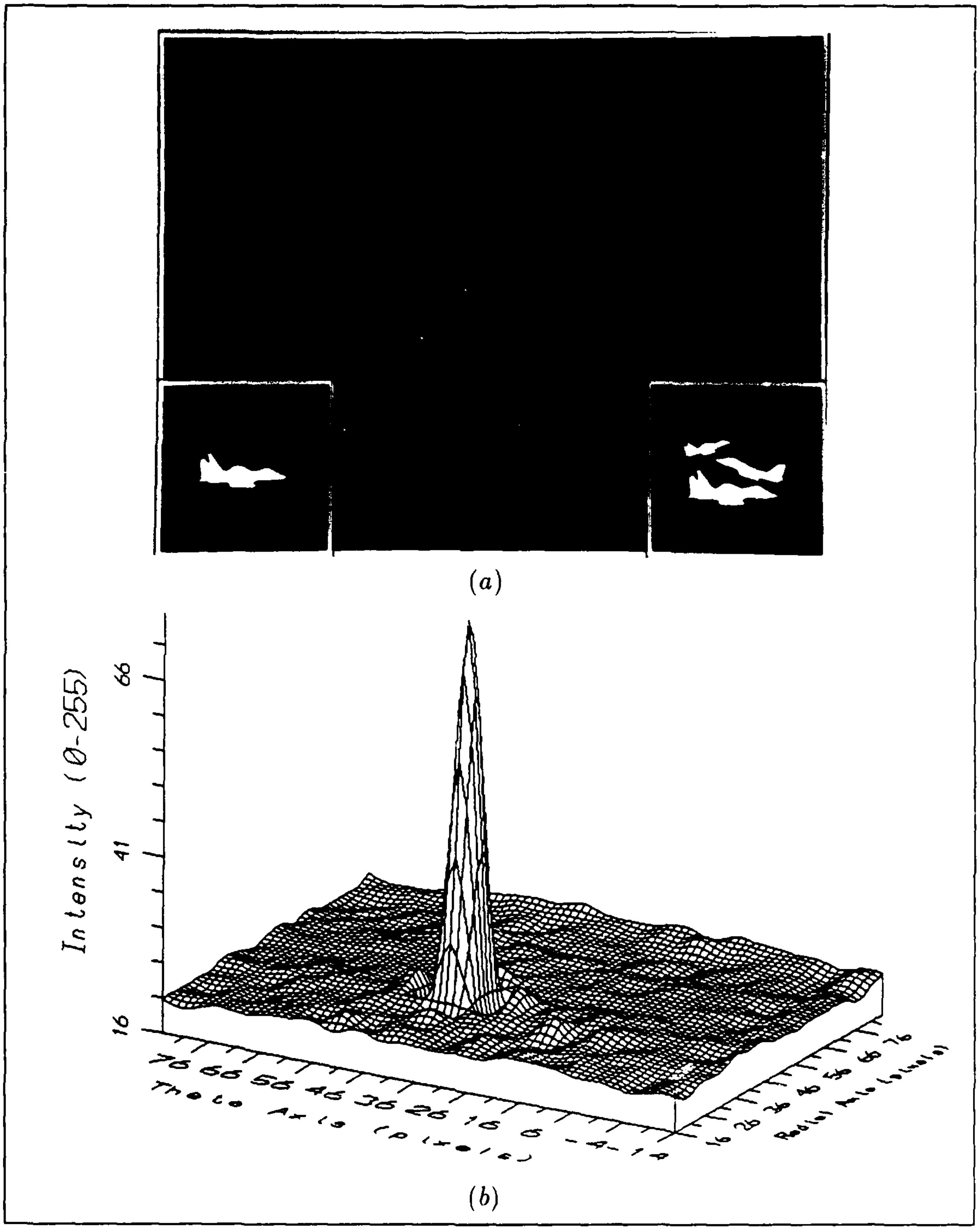
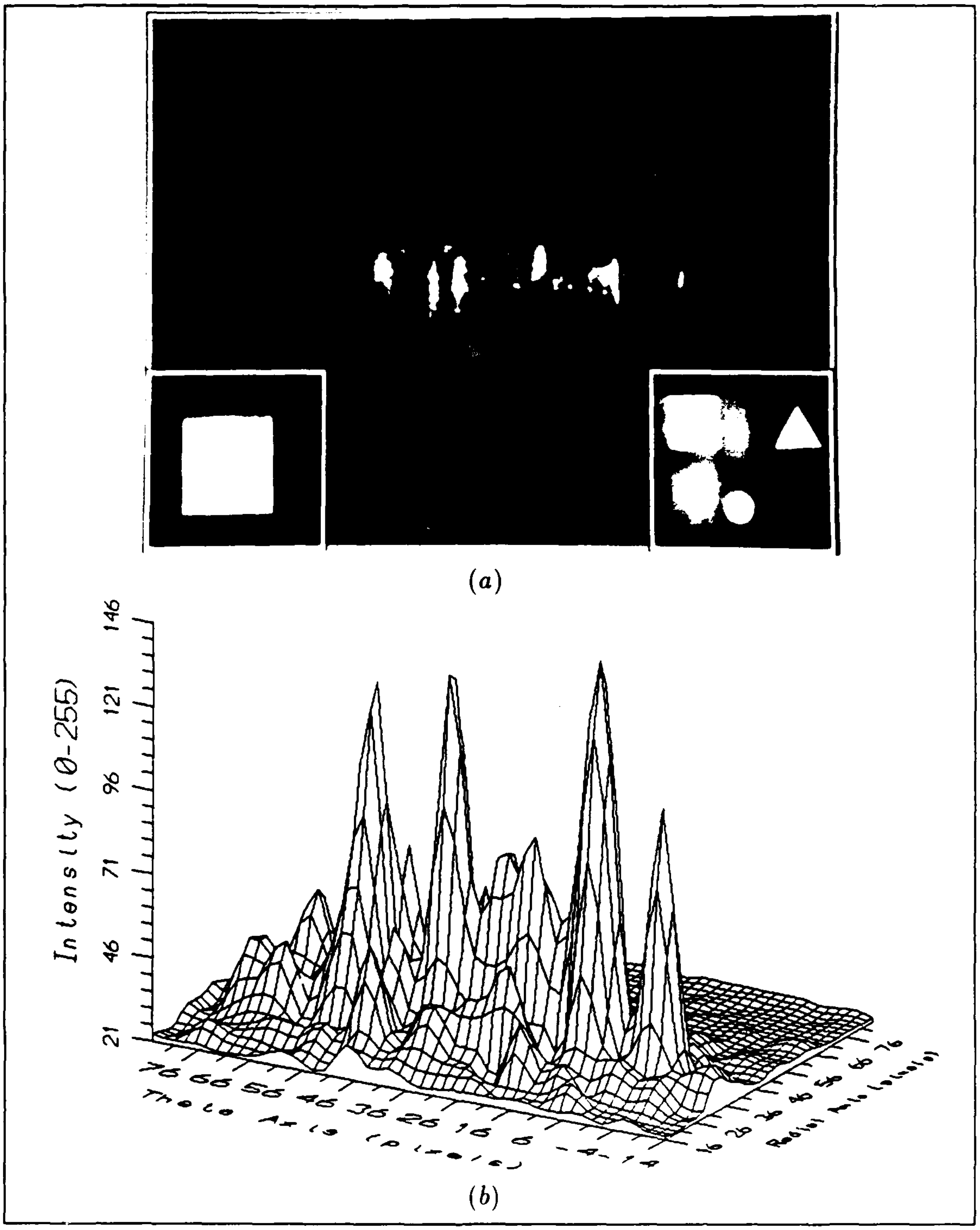


Figure 51. Optical correlation of the CTs of an F-15 and three F-15s where (a) is the intensity distribution and (b) is the 3-D plot. Peak corresponds to autocorrelation of two large F-15's (see Figure 56).



(a)

(b)

Figure 52. Optical correlation of the CTs of a square and a square, triangle, and circle where (a) is the intensity distribution and (b) is the 3-D plot. Note the large crosscorrelation peaks surrounding the autocorrelation peak at pixel location (36,36).

storing the output to disk.

In Figures 53–55, the correlations were of scenes that did not contain the target template. For these cases, high-intensity, autocorrelation-type peaks would not be expected. However, in Figure 53, the optical correlation of the CTs of a truck and an F-15 produces a very symmetric-looking, correlation peak. Figures 54 and 55, however, do show the more ragged nature that one would expect from a crosscorrelation.

In contrast to Figures 50–55, Figures 56 and 57 show optical correlations where the only input image was the template or the scaled version of the template. In comparing these eight figures, the following statements can be made:

1. Sometimes the autocorrelation peak is visually evident when the target is in the input scene. However, this is not always true.
2. Autocorrelation-type peaks are visually evident in some cases where the target is not in the input scene.
3. Scaled and rotated versions of targets provide very good autocorrelation results when they are the only images in the input scene.

These findings lead the researchers to believe that an intelligent peak-finder (such as a neural network) must be incorporated to distinguish the optically generated autocorrelation peaks from crosscorrelation peaks. Additionally, it is believed that going to the 4π space as opposed to the 2π space will remove much of the ambiguity from the identification process and provide much better resolution to the process as a whole.

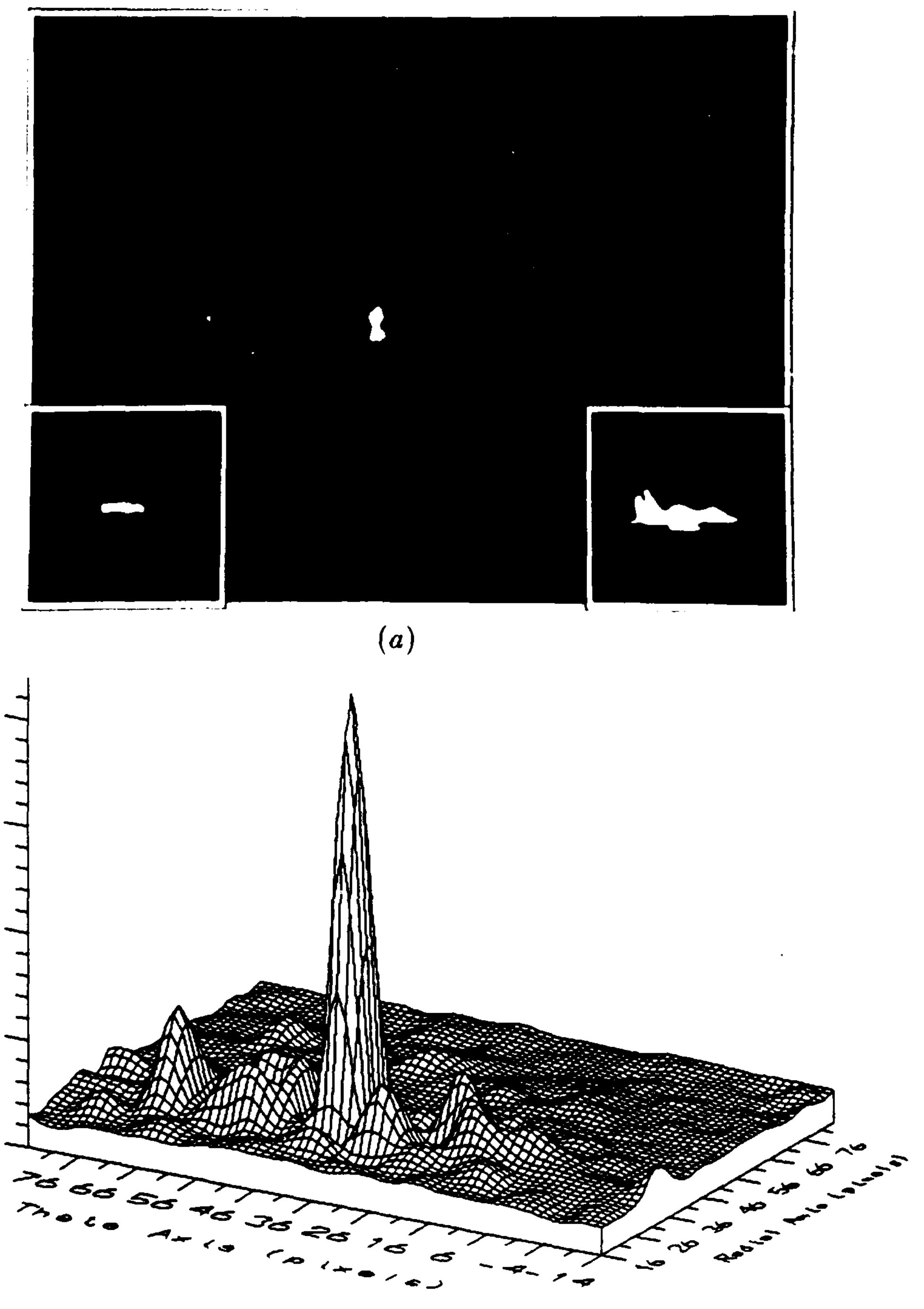
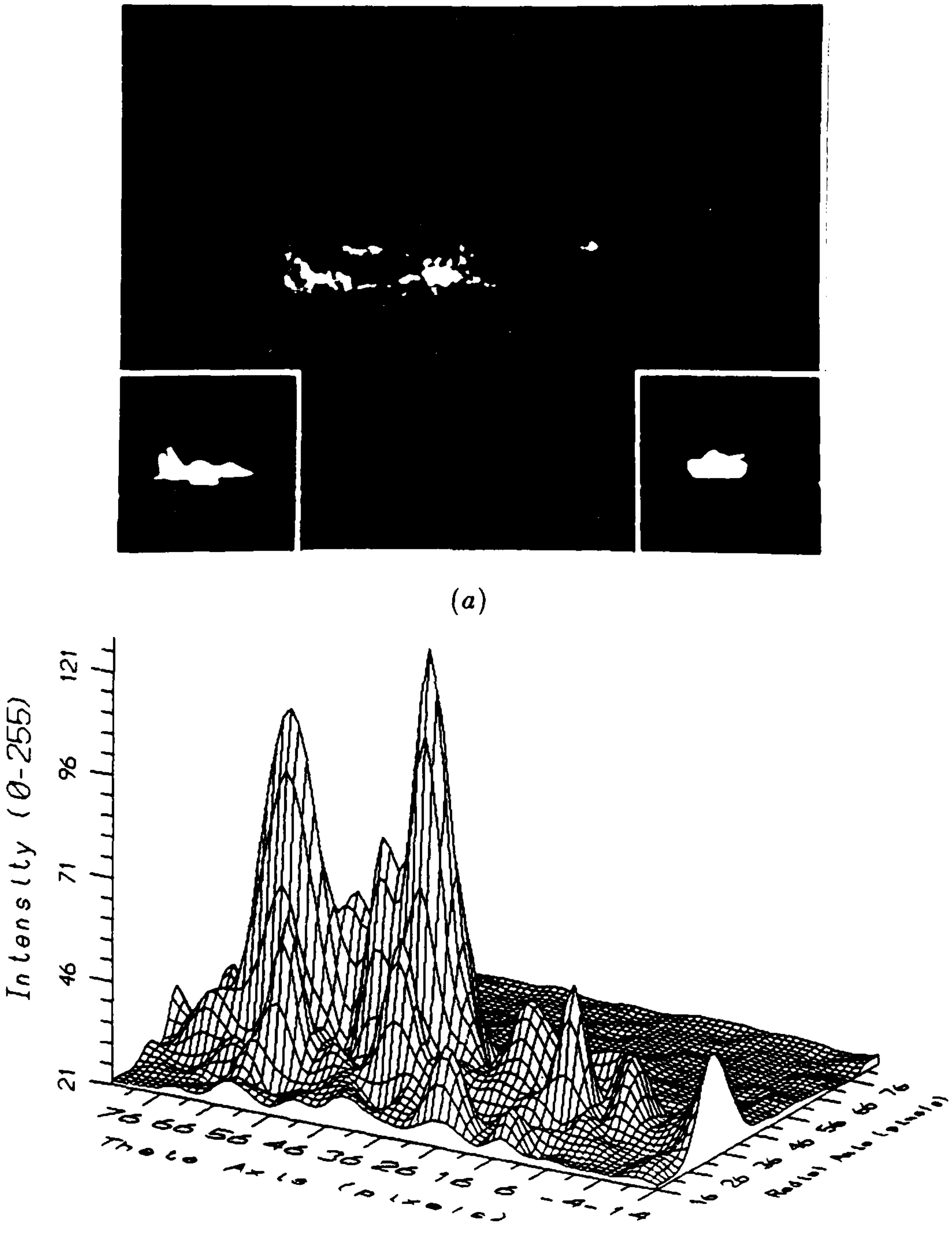


Figure 53. Optical correlation of the CTs of a truck and an F-15 where (a) is the intensity distribution and (b) is the 3-D plot. Note: Crosscorrelation peak could easily be confused as an autocorrelation.



(a)

(b)

Figure 54. Optical correlation of the CTs of an F-15 and a tank where (a) is the intensity distribution and (b) is the 3-D plot. Note the wide distribution of energy characteristic of a crosscorrelation.

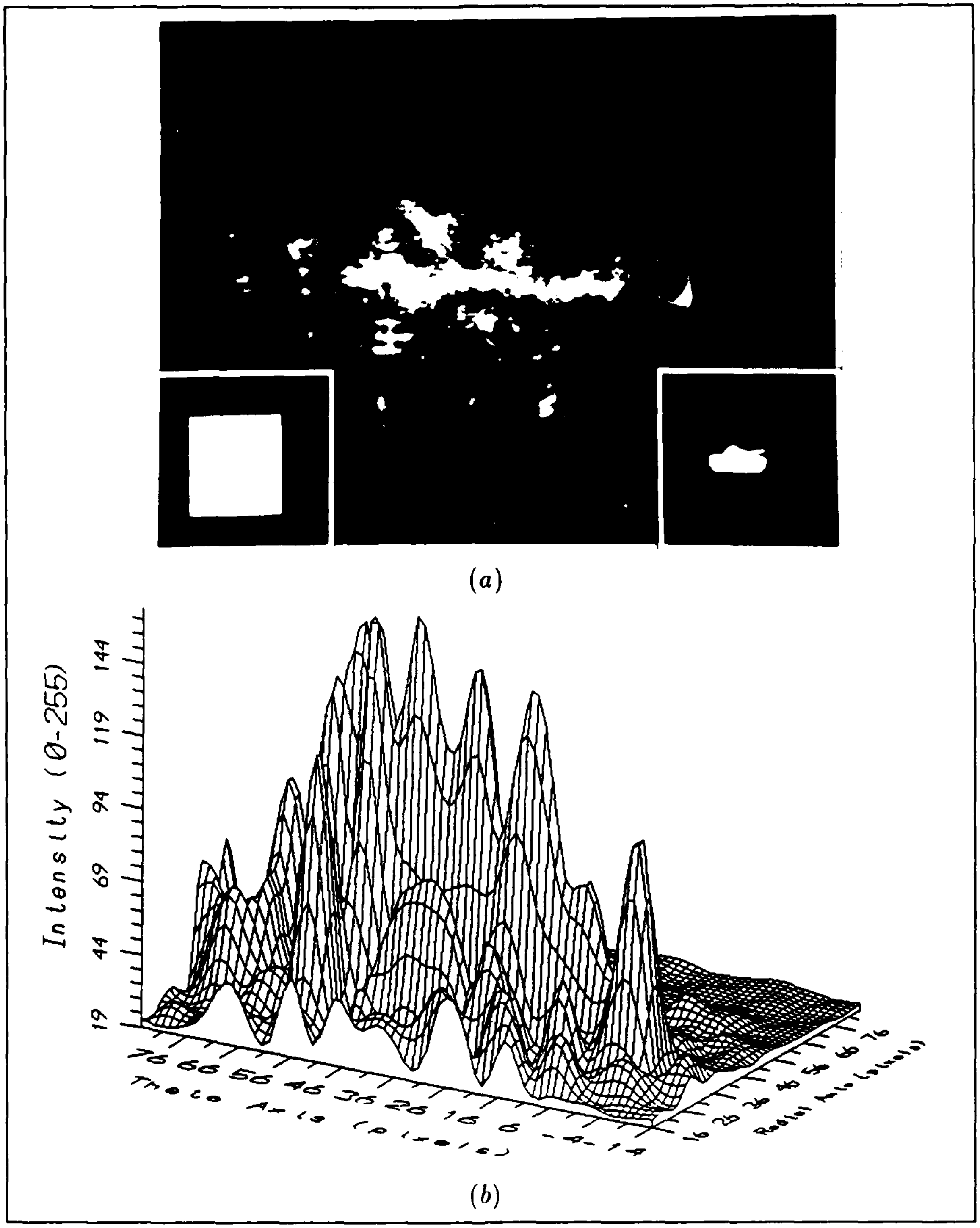
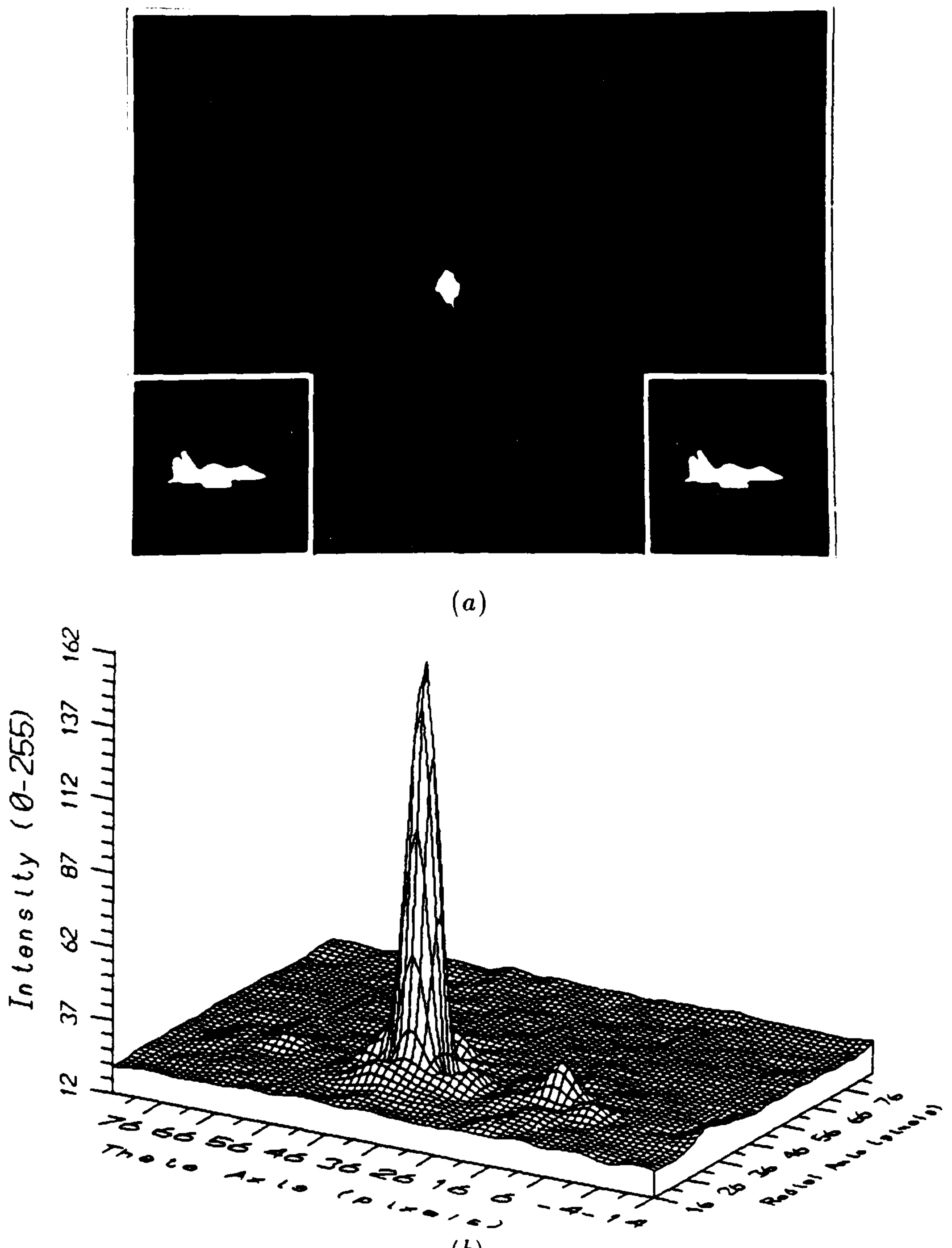


Figure 55. Optical correlation of the CTs of a square and a tank where (a) is the intensity distribution and (b) is the 3-D plot. Note the wide distribution of energy characteristic of a crosscorrelation.



(a)

(b)

Figure 56. Optical autocorrelation of the CT of an F-15 where (a) is the intensity distribution and (b) is the 3-D plot. Note the single, intense correlation peak. Its position is a reference for unity scale and 0° rotation.

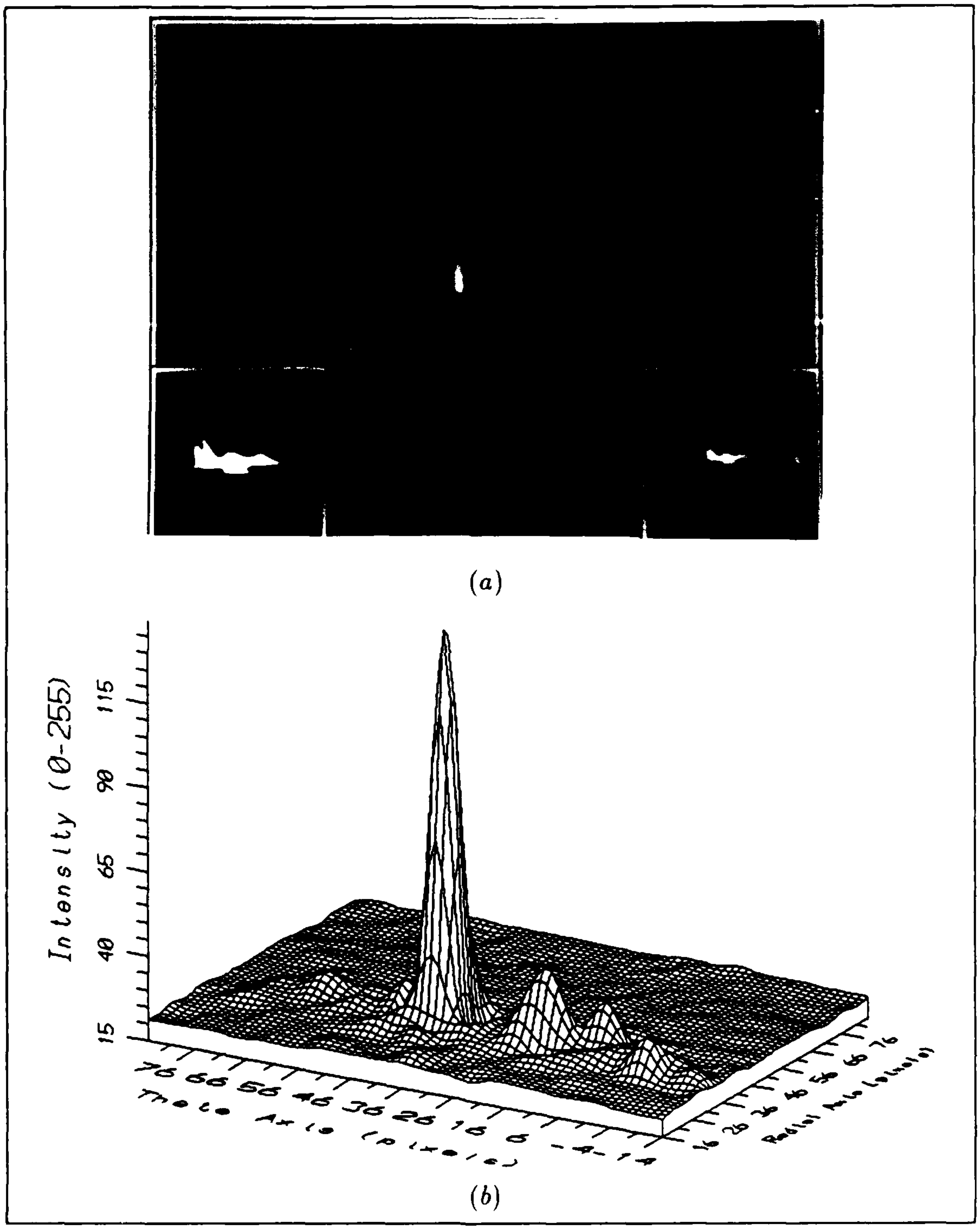


Figure 57. Optical correlation of the CTs of an F-15 and a scaled F-15 where (a) is the intensity distribution and (b) is the 3-D plot.

4.4 Summary

The results detailed in this chapter show that the $f_{\ln r} - f_\theta$ feature space can be a viable coordinate transform to be used in a position, scale, and rotation invariant pattern recognizer. Shifts in the position of autocorrelation peaks were shown to be predictable, which allows scale and rotation changes to be determined.

Various anomalies present in the current system were identified and discussed. Among these are the distortion of the CT caused by the CGH, the 4π versus 2π feature space, and the artifact produced by the sampling nature of the CGH. The following chapter discusses conclusions derived from the results and lists several suggested topics for further research.

V. Conclusions and Recommendations

5.1 Introduction

As stated in section 1.3, the scope of this thesis focused on demonstrating and evaluating portions of Miazza's optical design. In the next section of this chapter, conclusions are drawn, based on the results of this thesis effort. The last section of this chapter provides a myriad of recommended areas of study which could follow from this current research.

5.2 Conclusions

The goal of this research was to investigate the optical implementation of correlation techniques in the magnitude-squared Fourier transform $\ln r - \theta$ (FLRT) feature space. To accomplish this goal, a technique for recording the original input templates using rubylith templates was developed. In this technique, high resolution glass photographic plates were used to record the negative of the rubylith template. This process produced templates with vastly improved transmission efficiencies (in comparison to previously used Polaroid slides) and limited unwanted diffraction.

The next step required was to develop an improved method for generating the Fourier transforms of the input templates. This was accomplished by recording a circularly symmetric high-pass filter (HPF) on a high resolution plate, and then precisely positioning the HPF in the Fourier plane of a two-lens, Fourier transforming-imaging system. As a result of successfully eliminating the unwanted, low frequency components, system dynamic range was increased and the "blooming" effect previously seen on the CCD camera was eliminated. Also, for the first time, high-contrast black-and-white, instant slide film was used to record the intensity patterns of the Fourier transforms by imaging the transforms directly onto the film plane of a 35mm camera. This process produced the data needed for the next portion of the experiment—coordinate transforming.

To successfully perform the optical FLRT coordinate transformation (CT) of the magnitude-squared of the Fourier transform (*FT*), a new method of aligning and focusing the *FT* onto the computer generated hologram (CGH) had to be developed. This was accomplished by using a two-lens imaging system which allowed the *FT* to be imaged directly onto the CGH. Imaging onto the CGH, as opposed to the previous method of butting the *FT* against the CGH, vastly improved system alignment and focusing capabilities. At the same time, the two-lens system allowed the *FT* to be magnified by a factor of three to overcome a space bandwidth product problem associated with the resolution of the CGH. It was also shown that the CGH does behave in the expected fashion by giving a logarithmic shift along the vertical axis in response to scale changes of the input images and linear shifts along the horizontal axis in response to rotations of the input. As with the *FT*, the $|CT|^2$ was recorded using the high-contrast film and a 35mm camera.

With FLRT CTs in hand, a setup to optically generate Vander Lugt matched-filters had to be designed. A unique system was successfully developed that used a thermoplastic holographic camera to produce matched-filters in approximately one minute. The phase-relief holograms (matched-filters) produced by the thermoplastic camera were then used to produce successful optical correlations of the FLRT feature space for the first time.

In addition to developing an optical, FLRT correlator, this thesis also verified the optical results using digital simulations. CTs produced optically compared very favorably with the digital results, showing straight-line, vertical and horizontal shifts for scale and rotation changes, with the horizontal shifts being circular shifts. Also, both optical and digital CTs of vehicles tended to look very similar due to the rectangular nature of each. However, results showed that even with the similarities in appearance, the optical correlator could still differentiate between the FLRT feature spaces. Also, with the FLRT feature space, correlations produced optically exhibited predictable movements in response to scale and rotation changes, and autocorrelations versus crosscorrelations could be visually discerned in most cases. Thus, this research has shown that portions of the Kobel-Martin-Horev (KMH) algorithm can be implemented optically and are candidates for implementation in a real-time system.

5.3 Recommendations

This research has shown some promising results toward the goal of optically implementing a real-time, PSRI-based, pattern recognition system. It is recommended that research in this area continue, and to that end, the following list of specific topics is provided.

1. An important consideration concerning optical correlations of FLRT feature spaces is that a problem exists with trying to correlate the circular shift from rotations in the feature space using a linear correlation. Because of this, a 4π versus 2π feature space is required so exact autocorrelations can be made possible (see Figure 49). Additionally, going to the 4π space will eliminate the rotation ambiguity problem discussed earlier. Different methods of implementing a 4π-wide feature space should be investigated. Possibilities include using a spatial light modulator (SLM) to display two copies of a 2π-wide feature space positioned next to one another, developing a CGH that will automatically produce the 4π-wide space, or imaging two copies of the 2π-wide space onto the film plane of a 35mm camera to produce the wider feature space.
2. Spatial light modulators should be used to implement a real-time system. This can be accomplished by using the SLMs as input sources for binarized Fourier transform and coordinate transform intensity patterns, instead of using instant slide film to record those intensity patterns. Two electronically addressable SLMs are available for research work at this time.
3. Non-linear, barium titanate or lithium niobate crystals should be investigated as a possible mechanism for stripping the phase from a complex, input scene.

This may be possible because of the crystal's ability to produce uniform amplitude waves. If this can be done successfully, the entire KMH algorithm could be implemented in near real-time.

4. The computer generated hologram should be further characterized with respect to the SBWP problem. An approach here might be to produce several different sized copies of the same input template. And, in addition, collect data for a series of rotation changes—maybe in 10° increments. Also, the apparent nonlinear distortion produced by different regions of the CGH should be looked into.
5. Because a dynamic range problem did not allow sufficient resolution along the radial (vertical) axis in the FLRT space, improvement in the dynamic range should be investigated. One possibility is to use a cylindrical lens to magnify the correlation plane in the radial direction only. This will enhance the dynamic range by allowing shifts due to scale changes to be more readily detected.
6. Once a good set of 4π feature space correlations are obtained, some type of intelligent peak detector (maybe a neural network) should be trained to determine whether or not the correlation peaks are sufficiently developed to distinguish crosscorrelation peaks from autocorrelation peaks. Also, the correlation data obtained from this research could be analyzed with respect to the 4π data to determine the amount of improvement, if any, that is obtained.
7. Although this research concentrated on the $f_{\ln r} - f_\theta$ coordinate transform, other prospective transforms exist which could be used to produce feature spaces that are invariant to shifts, scale changes, and rotations. One of these is known as the Hough transform. These researchers believe that, with the methods used in this thesis, an optical Hough transform can be easily generated on the optics table. This should be pursued.
8. Another item that this research uncovered but did not investigate is the possibility of using the thermoplastic camera to produce phase-only filters. The

thermoplastic material produces a phase-relief hologram, as opposed to an amplitude hologram. With this in mind, it may be possible to illuminate the thermoplastic with the interference pattern of a phase function to record the phase function directly onto the film. If this works, it might be possible to generate a coordinate transform feature space of complex inputs in real-time.

9. Finally, since the feature space investigated does not handle out-of-plane rotations, it is recommended that further research should concentrate on applications where these rotations are unlikely or have been eliminated. Two examples are satellite imagery and synthetic aperture radar (SAR) data.

Appendix A. *DC-Block and Image Template Preparation*

A.1 Introduction

Based on the results of Mayo's work [12] and an unsuccessful attempt to use Polaroid slides, it became necessary to develop a method for producing high quality, high resolution dc-blocks (high-pass filters) and binary image templates. The dc-block used by Mayo was a small, black, ink spot on a clear glass slide. The ink spot was not symmetric (circular) and clearly not uniform in density. Thus, it did a poor job of uniformly attenuating the low frequency components of a Fourier transform and resulted in the production of non-symmetric, optical Fourier transforms.

Some of Mayo's image templates were very rough cutouts of geometric images and letters. These rough cutouts had the disadvantage of causing unwanted diffractions around the edges of the images. Additionally, an initial attempt in this thesis effort at using Polaroid instant slide film for recording template images failed miserably. The Polaroid slides greatly limited transmission and suffered from internal reflections which produced a multiple-image effect.

A.2 DC-Blocks

Why have dc-blocks in the first place? The answer is twofold. First, the information we want to detect, the outlines of targets in input scenes, produces high frequency terms in the frequency (spatial) domain. Thus, if we can eliminate the unwanted, low frequency energy, we can increase the relative strength of the higher frequencies. In other words, a kind of edge enhancement of the image. This edge enhancement makes it easier to "recognize" the target.

Secondly, because the low frequency component of a Fourier transform contains a delta function (mathematically speaking) derived from the average intensity of the input scene, this component is very intense. And, since a CCD camera was being used to detect the intensity when recording Fourier transforms, a conscious effort had to be made to eliminate an affect known as "blooming." If the CCD array is subjected to a beam intense enough to saturate the array elements, electric charges collected in the elements will "overflow" into surrounding elements. Thus, a "blooming" affect. To prevent this, an effort had to be made to ensure saturation did not occur. This could have been accomplished by attenuating the entire input beam, including the dc portion. However, this would have severely limited the dynamic range of our system, relative to the high frequency components, since the intensity of the desirable frequencies would have been attenuated to an unacceptably low level. Therefore, blocking the dc component allowed the amount of attenuation required to prevent "blooming" to be reduced while, at the same time, enhanced the dynamic range of our system.

A.3 Preparation

To solve the problem of preparing high quality dc-blocks and binary image templates, this thesis effort used Kodak, high resolution, glass plates (2"x2" Type 1A ultra-flat photographic plates) . The plates used are the same type previously used by Mayo to reduce and record the coordinate transforming, computer generated hologram (see Appendix B). These plates are available through GSA Contract GS-00F-77352, modification 61, dated February 28, 1987. Facilities for exposing and developing the glass plates were available in the Cooperative Electronics Materials Process Lab, room 1065 of building 125. Figure 58 shows the optical setup of the Dekacon camera used to reduce and record the dc-blocks and binary templates.

A.3.1 The Dekacon The Dekacon camera system consists of four main subsections (see Figure 58. First, the object screen is a four foot by four foot apparatus which contains a green light source to provide the illumination required for film exposure. It is also a rigid support structure used to hold the transparency being photoreduced.

Second, the camera support table is a heavy, rigid steel device which has been calibrated to allow precise positioning of the front and rear boxes. The rigidity and weight of the table also provide considerable stability for eliminating unwanted vibrations (noise) which could affect the quality and resolution of the photoreduction.

Next, the front box is the mounting mechanism for any one of several available lenses. A 35mm lens is available, as are a 12-inch Wray, a 5-inch Wray , and a 3-inch Wray. The 3-inch and 5-inch Wrays were used in this thesis effort for preparing 10x and 20x reductions of image templates.

The last subsection is the rear box which provides the mounting mechanism for the high resolution film plates. The rear box, in conjunction with the front box, can be moved along the calibrated table to position it for the desired photoreduction power.

A.3.2 The Procedure The first step is to position the front and rear boxes according to the amount of reduction desired. A manual available in room 1065 details the required box settings for the various reductions for the different lenses. Once the boxes are in place, the Dekacon has to be focused for the particular reduction desired before any photoreduction can be accomplished. This is done by first mounting a scratched glass plate in the film plane with the scratches facing toward the object screen. Now, with the object to be focused mounted on the object screen, a microscope objective is used to bring both the object and the scratches simultaneously into focus. At this point, the camera is ready to do a photoreduction.

With the template in position on the object screen and the camera focused for the desired reduction, an unexposed high resolution plate is inserted into the film holder portion of the rear box. This must be accomplished with all lights, except red, extinguished. Once the unexposed plate is in position, the green light of the object screen is illuminated and the plate is exposed. This research has found that an exposure time of four to five minutes provides enough photons to successfully expose the film plate.

After exposure, the plate is removed from the rear box (again with the lights off) and taken to a dark room for development. In this research, the following times were used: development time—2 minutes, stop time—20 seconds, and fixer time—1 minute. Depending upon the age of the film plates and the age of the developer, the exposure time and development time may need to be varied until the best quality is obtained. One method of testing the developer is to put an unexposed plate in the developer and determine the time required for the plate to start to become cloudy. That time period should then be used for the development time.

A.3.3 Templates The template used for making the dc blocks was a common flat washer. The washer was painted black in an effort to reduce any stray reflections and then mounted onto the object screen. By changing lenses and reductions powers, several different sized black dots were produced. These dots would then be used as the dc blocks.

To produce the image templates, red Rubylith was manually cut into the desired shapes. These Rubylith cutouts were then mounted on the object screen and photoreduced. The reason Rubylith was used is that it blocks the green light from the object screen while, at the same time, was easy to cut into the desired shapes.

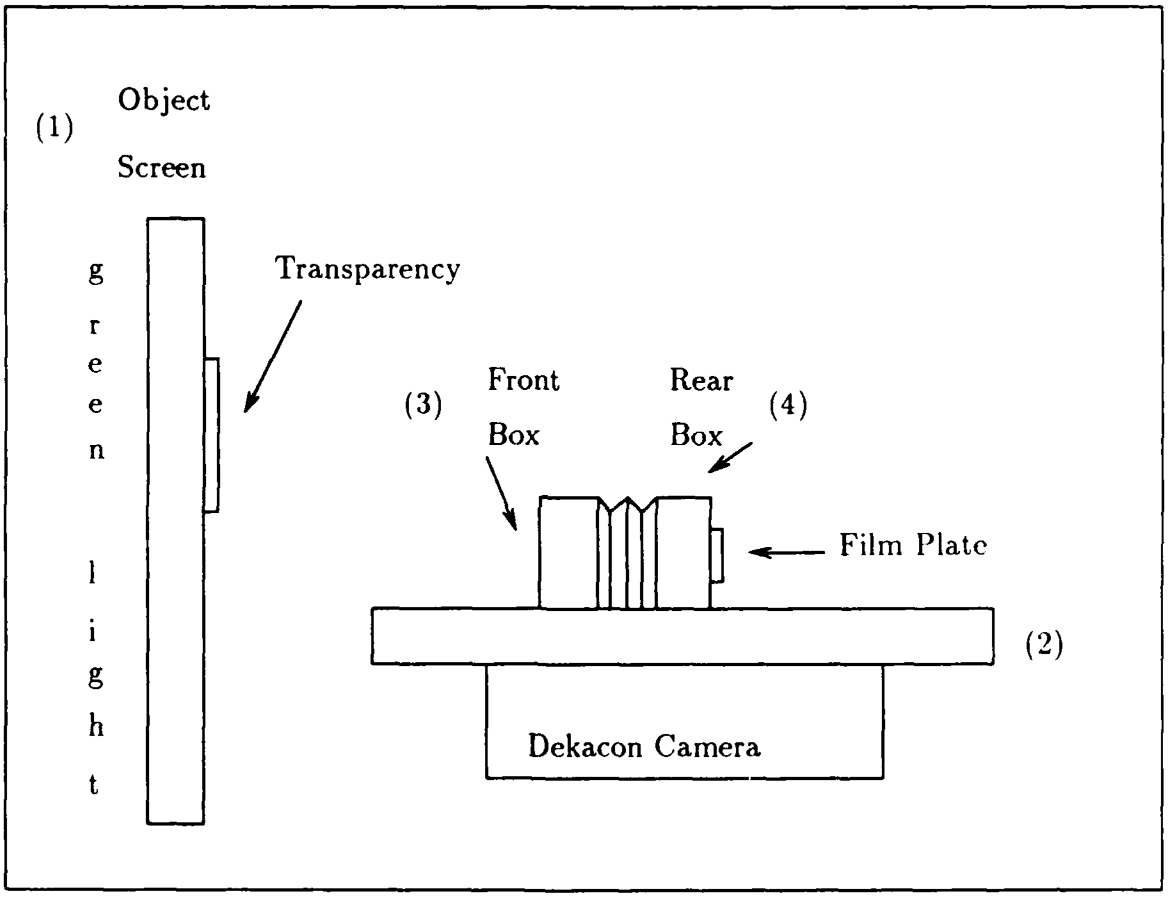


Figure 58. Dekacon Camera Setup, Rm 1065 of Bldg 125. Note: Used to photoreduce image templates, dc blocks, and CGHs.

Appendix B. *The Computer Generated Hologram*

This appendix details the methodology used for producing a computer generated interferogram. The interferogram of interest is one that performs an optical $\ln r - \theta$ coordinate transformation. Much of the following discussion is adapted from the work of Mayo [12].

B.1 Generation

The computer generated hologram (CGH) is a binary fringe pattern which has been produced (printed) by a laser printer and transferred, via photoreduction, to a high resolution glass plate. When illuminated with coherent light, the CGH will diffract the light in such a way that many diffracted orders are produced. To ensure that the first diffracted order, which contains the coordinate transform, is separated from the higher-order diffractions, a sufficiently high carrier frequency must be used. The following equation must be satisfied for the carrier frequency α to ensure no overlap of diffracted orders occurs

$$\alpha > \left[\frac{3}{\lambda f_L}\right] \ln \left[\left(\frac{1}{2}x_{max}\right)^2 + (1/2y_{max})^2\right]^{\frac{1}{2}} \tag{23}$$

Where x_{max} and y_{max} are input dimensions, λ is the wavelength of light used, and f_L is the focal length of the transforming lens used in conjunction with the CGH.

For this particular work, the final dimensions of the CGH were to be $x_{max} = y_{max} = 10$mm, the wavelength of light was $\lambda = 632.8$nm (Helium-Neon laser source), and the focal length was $f_L = 300$mm. With these parameter values, the value for α necessary to solve the equation is 30.9 lines per millimeter. An α of 35 lines per millimeter was chosen to meet the requirement.

To create the CGH, the binary fringes of the transmittance function were printed on a laser printer and then optically reduced using the technique discussed in

Appendix A. For the ln $r-\theta$ coordinate transform CGH, the transmittance function obtained by interfering the required phase function given by

$$\phi(x,y) = \left[\frac{2\pi}{\lambda f_L}\right]\left[\ln(x^2+y_2)^{\frac{1}{2}} - y\arctan(y/x) - x\right] \tag{24}$$

with a carrier, α, is given by

$$t(x,y) = 0.5\left[1+\cos(2\pi\alpha x - \phi(x,y))\right] \tag{25}$$

The transmittance function then has fringes (maxima) where the argument of the cosine satisfies

$$2\pi\alpha x - \phi(x,y) = 2\pi n_k \tag{26}$$

for integer values of n_k, where each value of k corresponds to a different fringe.

Equation 26 is then used to generate a 128 by 128 array of numbers of the maximums at evenly spaced intervals. The Metalib program on the AFIT/ENP Sun workstation is used to draw the contour plot based on these numbers. This plot is the CGH. An example is shown in Figure 59.

B.2 The Metalib Program

As previously mentioned, a program called Metalib was used to plot the CGH. In particular, a routine of Metalib called RCONTR was used. This routine is shown below [5:49–55].

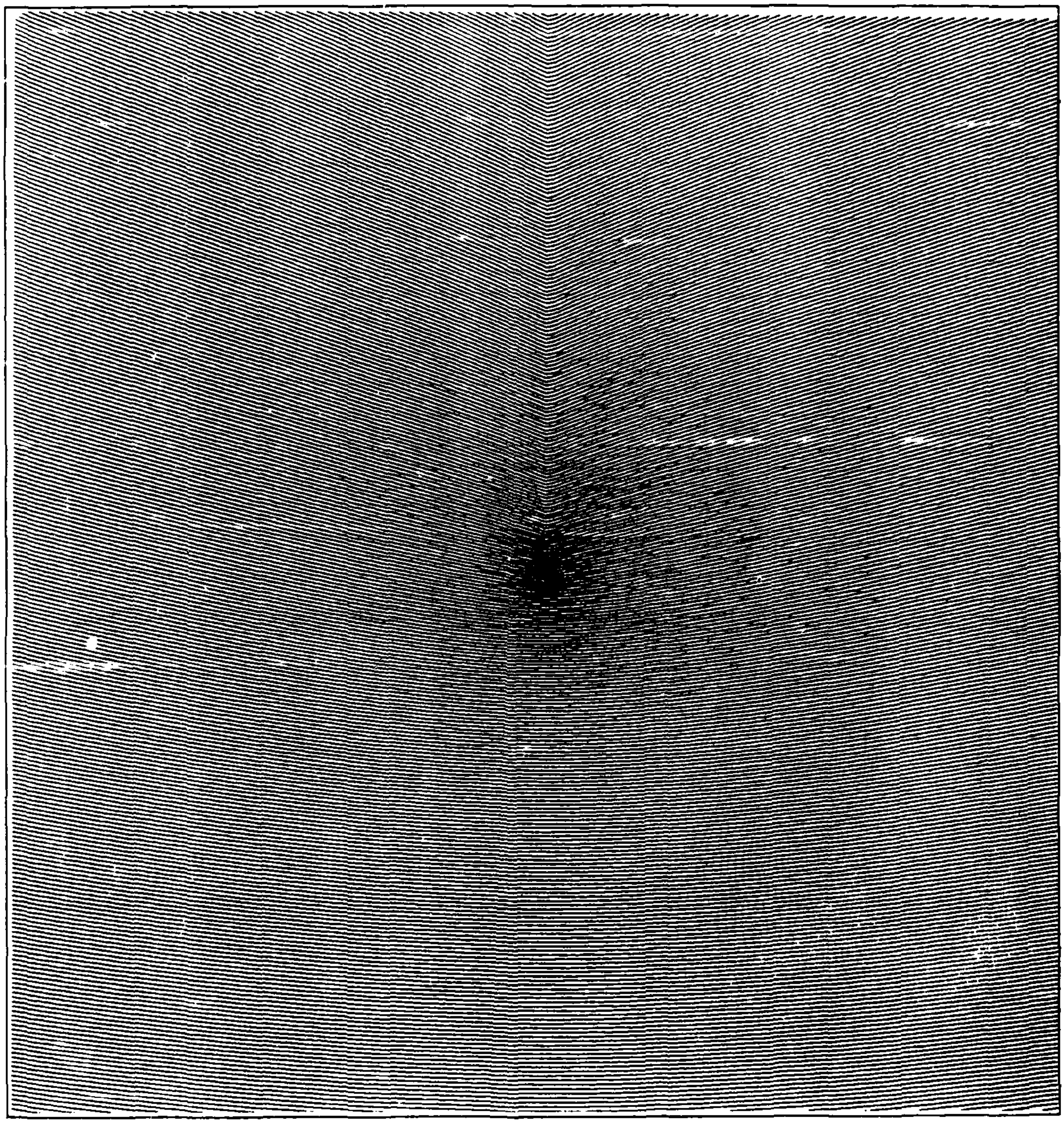


Figure 59. Laser printer plot of a CGH with $\lambda = 632.8\text{nm}$ and $f_L = 300\text{mm}$. (Note: photoreduction by 20 to 1 will give an $\alpha > 35$ lines/mm).

```
C
C ********* LN R.THETA INTERFEROGRAM **************
C
       real     SURFAC(128,128), X, Y, ALPHA, LAMDA, FOCAL, TEMP, PI,
     + INCX, INCY, CONTV(2), KS, Xcoord(128), Ycoord(128)
integer ICTRS
C
ICTRS = 0
KS = -400.0  'not needed--returns number of lines plotted
CONTV(1) = -350.0  'set to minimum value in array
        CONTV(2)        = .6      'delta value
C
X = -2.51   'minimum dimension of CGH
Y = -2.51   'minimum dimension of CGH
ALPHA = 45.0   'Angle of plane reference
FOCAL = 300   'focal length of associated lens
LAMDA = 0.0006328   'wavelength
INCX = 5.0 / 128.0 'step increment of x
INCY = 5.0 / 128.0 'step increment of y
PI = 3.1415926
C
do 10 j = 1, 128, 1                'this loop generates data
   do 20 i = 1, 128, 1
      if ( (X.lt.0) .and. (Y.gt.0) ) then
 TEMP = X*0.5*alog(X*X+Y*Y)-Y*(PI+atan(Y/X))-X
      else if ( (X.lt.0) .and. (Y.lt.0) ) then
 TEMP = X*0.5*alog(X*X+Y*Y)+Y*(PI-atan(Y/X))-X
      else
 TEMP = X*0.5*alog(X*X+Y*Y)-Y*atan(Y/X)-X
      end if
      SURFAC(i,j) = ALPHA*X - (1/(LAMDA*FOCAL))*TEMP
      Xcoord(i) = X
      X = X + INCX
 20    continue
   Ycoord(j) = Y
   X = -2.51
   Y = Y + INCY
 10 continue
C
C **** METALIB Subroutines ***  (described in manual)
C
call setplot( "lnplot128", 7, 0, .false.)
```

```
C
call mapxy( -2.51, 2.49, -2.51, 2.49, 0.0, 0.7, 0.0, 0.9, 0, 1 )
C
      call rcontr( ICTRS, CONTV, KS, SURFAC, 128, Xcoord, 1, 128, 1,
     +      Ycoord, 1, 128, 1, .false. )
C
call endplot
C
      end
```

The function of the METALIB subroutines SETPLOT, MAPXY and ENDPLOT are explicitly explained in the METALIB manual. In the RCONTR subroutine, the role of ICTRS, CONTV(1), CONTV(2), and KS can be confusing. With ICTRS = 0 all contours of the form $CONTV(1) + m * CONTV(2)$, m determined by METALIB, are drawn. METALIB assumes CONTV(1) contains a starting value and CONTV(2) a delta value. It is computationally economical to set $CONTV(1) = \min SURFAC(X,Y)$, however is may be set to any value below $\min SURFAC(X,Y)$ but not above. CONTV(2) controlls the density of the plot. METALIB increments on CONTV(2) to draw contours by interpolating between values given in SURFAC(X,Y). CONTV(2) must be adjusted to fit the specific conditions of wavelength, alpha, focal length, and plot dimensions. CONTV(2) = .6 gave the best results for the parameters used.

Appendix C. *Sight-Mod Spatial Light Modulator*

During this thesis effort, AFIT's Electrical Engineering Optics Laboratory acquired two 128 by 128 pixel Semetex Sight-Mod binary spatial light modulators (SLM). These devices are very similar to the Litton SLM investigated by Mayo [12]. Both are electronically addressable, magneto-optical devices which modulate information on a linearly-polarized light beam by individually rotating the polarization of light incident on each individual pixel. The polarization is rotated either clockwise or counterclockwise depending on how the pixel was addressed. A polarizer at the output is positioned to pass one polarization state and block the other.

The main difference between the SLM investigated by Mayo and the Sight-Mod is the computer interface's addressing scheme used to electronically address the 128 by 128 array. Mayo's Litton device was originally designed to interface with an Apple II computer. Mayo converted the interface and software so the Litton SLM could be addressed with a Zenith 248 (IBM compatible) computer. The Litton SLM was designed to address only one pixel at a time. The Sight-Mod was designed to interface with an IBM AT (or compatible) computer addressing 8 pixels at a time. Consequently, the software developed by Mayo to drive the Litton device is incompatible with the Sight-Mod.

The Sight-Mod has provided source code (written in the C and Basic programming languages) for demonstration routines for testing the device. Eight pixels are addressed at a time by writing a byte (8 bits) to an assigned segment in the computer's memory. The address segment is selected using the dip switches located on the Sight-Mod computer interface board. Details on changing the segment address and the memory addressing scheme can be found in the "Sight-Mod Development System Operations Manual" [17]. D000 was the address segment used during this thesis research. For the 128 by 128 array, an image is displayed by individually addressing memory locations D000–D7FF. Each memory location addresses 8 consecutive pixels in a column of the array. With this format, a column of the SLM (128 pixels) can be written to by addressing only 16 bytes of memory. Semetex calls the eight rows addressed by one byte a "byte row". For example, if an array has a pixel location (r,c) where r is rows 0–127 and c is columns 0–127, memory location D000 would correspond to eight pixels (0,0)-(7,0) in column 0. The binary state of each bit of the eight-bit word written to a memory location determines the on or off state of the pixels addressed. For a byte written to D000, a one in the least significant bit (bit 0) will nucleate the (0,0) pixel, and a one in the next bit will nucleate pixel (1,0). Nucleation is an intermediate state of a pixel that is being set. A pixel is completely set after the magnetic coil surrounding the array is excited. This is accomplished by addressing memory location D801 after all pixels have been addressed, thereby completing the writing process. Note that reading the most significant bit of memory location D800 will indicate if the SLM is busy (it will be a 1) or ready (it will be a 0). In addition, the entire array can be erased by addressing D800.

The SLM has considerable potential as a tool for demonstrating optical architecture proposed for real-time pattern recognition. The applications range from a incoherent-to-coherent image converter to removal of the phase from a complex wavefront. For this research, the SLM was used to modulate intensity distributions of complex wavefronts detected and stored with a CCD camera and the AT&T TARGA framegrabber. The computer source code used to redisplay TARGA images is listed below. The program was written and compiled using Turbo C. This program takes a 512 by 400 pixel, 256 grey level, TARGA format image and reduces it to 128 by 100 pixels by averaging and binarizes it by thresholding. The scaled, binarized image is then written to the SLM.

```
/*************************************************************/
/*      SLM_DISPLAY_TARGA                                    */
/*                                                           */
/*     Written by J. Thomas Walrond and Timothy Childress    */
/*     Program displays a TARGA image file on the Sight-Mod  */
/*     Spatial Light Modulator. First the image is reduced   */
/*     from 512 by 400 pixels to 128 by 100 pixels by        */
/*     averaging. Then the image is binarized by thresholding.*/
/*     Then the image is reformatted and written to the Sight-*/
/*     Mod.                                                  */
/*************************************************************/

#include "stdio.h"
#include "dos.h"
#include "math.h"

int screen=0xb000, value=0xf0, val;
unsigned temp [4] [512], avgtemp [128] [128], temp1;
int i,j,jj,ji,x,y,r,avgi,l,favg, bt,nr;
char ctemp, cval,slm [16] [128];
float avg,sum,ftemp, jf, thrshld=50;
int c,busy;

FILE *fr1;

main()
{       pokeb(screen, 0x800, 0xff);/*clear slm*/
        for(busy=0x80; busy== 0x80; )
        busy = 0x80 & peekb(screen,0x800);
/*Retrieve a a Targa CCD file (512x400), reduce to 128x100 by blurring.*/

        fr1=fopen("I13_fc.tga","rb");/*get ccd file*/
      for(r=0; r<100;r++)/*retrive data sequentially into 4x512array*/
{
for(x=0; x<4; x++)
{
for(y=0; y<512; y++)
{
                                ctemp=getc(fr1);
temp1=ctemp;
                                if(temp1>255)temp1=temp1-65280;
temp [x] [y]= temp1;
/*printf("%u \n", temp [x] [y]);
printf("%u",x);
printf("%u \n",y);
printf("%u \n", ctemp [x] [y]);*/
}
}
                for(j=0; j<512; j+=4){/*reduce 4x512 array to 1x128 by avg*/
    sum=0;
    for(jj=j; jj<j+4; jj++){
for(i=0; i<4; i++){
     ftemp=temp [i] [jj];
     sum=sum + ftemp;
         }
    }
    avg=ceil((sum/16)-0.5);
    avgi=avg;
    jf=j;
                    jf=jf/4;
    ji=jf;
    avgtemp [r] [ji] =avgi;
    /*printf("%u",avgi); */
       }
        }
fclose(fr1);
/*binarize pixels based on threshold*/
        for(x=0; x<128; x++){
       for(y=0; y<128; y++){
   /*printf("%u \n", avgtemp [x] [y]);*/
   if(avgtemp [x] [y] < thrshld)avgtemp [x] [y] = 1;
   else avgtemp [x] [y] =0;
   /*printf("%u \n", avgtemp [x] [y]);*/
       }
       /*printf("%u \n", avgtemp [x] [y]);
       printf("%u", x);
       printf("%u", y);*/
}
/*convert 8 pixels into a eight bit character for addressing the SLM*/
for(c=0; c<128;c++){
    nr=0;
    for(r=0; r<96; r+=8){
bt=r;
```

```
val=0;
if(avgtemp [bt] [c] == 1)val=val+1;
bt++;
if(avgtemp [bt] [c] == 1)val=val+2;
bt++;
if(avgtemp [bt] [c] == 1)val=val+4;
bt++;
if(avgtemp [bt] [c] == 1)val=val+8;
bt++;
if(avgtemp [bt] [c] == 1)val=val+16;
bt++;
if(avgtemp [bt] [c] == 1)val=val+32;
bt++;
if(avgtemp [bt] [c] == 1)val=val+64;
bt++;
if(avgtemp [bt] [c] == 1)val=val+128;
bt++;
/*printf("%u \n", val);*/
cval=val;
/*if(cval>128) temp1=temp1-65280;*/
slm [nr] [c]=cval;

/*printf("%u \n", slm [nr] [c]);*/
nr++;

}
  }
/*  nr=13;
  for(c=0; c<128; c++){
      for(r=96; r<128; r+=8){
                val=0;
cval=val;
                slm [r] [c] = cval;
      }
  }     */

   r=0;        /* fclose(fr1); */
   for(x=0; x<16; x++){
for(y=0; y<128; y++){
      /*printf("%u \n", slm [x] [y]);*/
               pokeb(screen, r,slm [x] [y]);
      r++;
      }
  }
  pokeb(screen,0x801,0xff);
 }
```

It also should be noted that TARGA has a software package called "TARGA Tools" that could be very useful in interfacing the CCD camera with the Sight-Mod in a real-time fashion. These programs have been written in C and compiled using the Microsoft C-86 compiler. TARGA tools enables a programmer to remotely grab an image into the TARGA board's external memory where it can accessed for further processing, such as displaying on the Sight-Mod. The TARGA board is capable of grabbing an image into memory in 1/30th of a second. The Sight-Mod is capable of displaying a frame as fast as 1/300th of a second. Therefore, the only other limiting factors are the memory access time for reading the TARGA memory and the computational time for reducing, binarizing, and reformatting the data for display on the Sight-Mod.

Appendix D. *CCD Camera Image Format*

D.1 CCD Images

To save resultant images during this research, a Sony, charged coupled device (CCD) camera was used in conjunction with an AT&T Truevision Raster Graphics Adapter (TARGA 8). The TARGA 8 was installed in a Zenith 248 (IBM AT compatible) computer. The TARGA 8 board captures a full NTSC frame in 1/30th of a second. The captured image is digitized into a 512 by 400 pixel array using a 256-level grey scale. The TARGA 8 then displays the digitized image on a composite monochrome monitor.

Using the True_art software package provided with the TARGA 8, images can be captured and saved as a computer file for image processing or redisplay. Digitized images are stored as 8-bit, unsigned integer pixels in a 512 by 400 row-major format array. The array is stored in a sequential file where each record is a row of imagery.

An image file stored on a MS DOS formatted 5 1/4 inch disk can be transfered to the VAX I780A using the binary file transfer protocol (FTP) on the ETHERNET. The file transfer can be accomplished using the following procedure:

1. Using an IBM compatible computer connected to the ETHERNET, type "ftp <return>" to enter the file transfer protocol program.
2. Type "binary <return>" to indicate that a binary file is being transferred.
3. Type "open imv2a <return>". Then type "user <return>" at the prompt for username, and type "<return>" at the prompt for a password. The IMV2A is a gateway for file transfer on the ETHERNET to the DECnet.
4. Type "put filename1 filename2 <return>" where filename1 is the name of the file you are transfering from the the disk, and filename2 is the name the file will have in the USER directory of the IMV2A. Filename1 and filename2 can be the same name but do not have to be.

5. Once the file transfer is complete, the FTP can be exited by typing either "bye <return>" or "quit <return>."
6. The file transfer can now be completed by logging on to the VAX I780A and using the copy command to copy the file from the IMV2A. Once the copy is complete, delete "filename2" from the IMV2A USER directory.

CCD images were transfered to the VAX I780A for digital processing and future viewing. The two computer programs which utilized these images the most were "Executive," developed by Kobel and Martin to digitally implement the pattern recognition algorithm proposed by Horev [9], and "Hologram_ Menu," a digital simulation of an optical phase-only filter designed to perform $\ln r - \theta$ coordinate transforms. The required image data formats for both programs were different than the CCD image format. Two programs were written in ADA to convert the CCD images into a format useable by "Executive" and "Hologram_ Menu." The first program is called "CCD_to_256_USB_Format." This program takes a CCD image file and reduces the image from a 512 by 400 array to a 256 by 256 unsigned byte file. This program also has the option to binarize the image. "Executive" and "Hologram_ Menu" are both designed to process 256 by 256 image arrays.

The second program is called "Format_256_image." This program gives the user the option of converting a 256 by 256 unsigned byte image array into the pseudo signed byte format used in "Hologram_ Menu", or to the complex number format used in "Executive." This program can also convert image files from "Hologram_ Menu" and "Executive" back to the unsigned byte format. The unsigned byte format is used by the microVAX Fortran display program "Display_Images" which was adapted from a program of the same name written by Mike Roggemann.

The program code for "CCD_to_256_USB_Format" and "Format_256_image" and supporting software packages are listed below:

```
---------------------------------------------------------------------------
---------------------------------------------------------------------------
--          Procedure CCD_TO_256_USB_FORMAT                              --
--                                                                       --
--    Written by J. Thomas Walrond and Timothy Childress                 --
---------------------------------------------------------------------------
---------------------------------------------------------------------------

-- Description: Procedure takes an image (400 by 512 pixel) stored using
-- the AT&T TATGA frame grabbing system and reduces the image down to a
-- size useable by the procedures Executive and Hologram Menu (256 by
-- 256 pixels).  The procedure gives two options in reducing the image
-- size:  sampling and cropping.  A binarizing option is available for
-- cleaning up binary templates.  The binarizing threshold value is 128.

with Data_types; use Data_types;
with File_io; use File_io;
with text_io; use text_io;
with float_text_io; use float_text_io;
with integer_text_io;   use integer_text_io;
with IMAGE_TOOLS;       use IMAGE_TOOLS;


procedure CCD_TO_256_USB_FORMAT  is

   n1, n:  string(1..80);
   l1, l:  integer;
      im:  im_ar(1..401,1..512);
    exim:  im_ar(1..512,1..512);
fim, sim:  im_ar(1..256,1..256);
 answer1:  integer;
 answer2:  character;

begin
  MAIN_LOOP:
  loop
   put("Enter file name> ");
   get_line(n, l);
   get_image(n(1..l), l, im);
   put("CCD TO 256 BY 256 IMAGE FORMATING OPTIONS:");
   new_line;
   put("   1.  Expand ccd image to 512 by 512 with zero fill");
   new_line;
   put("       and sample to 256 by 256");
   new_line;
   new_line;
   put("   2.  Crop ccd image down to 256 by 256");
   new_line;
   loop
     put("enter option (1 or 2)>");
     get(answer1); skip_line;

     case answer1 is
        when 1 =>     EXPAND_CCD_IMAGE(im, exim);
                      SAMPLE_IMAGE(EXIM, sim);
                      exit;
        when 2 =>     CROP_CCD_IMAGE(im, sim); exit;
        when others => null;
     end case;
   end loop;
   loop
     put("Binarize image (y or n)?>");
     get(answer2); skip_line;
     case answer2 is
        when 'y' | 'Y' =>   BINARIZE_IMAGE(sim, fim); exit;
                            Put("binary complete");new_line;
        when 'N' | 'n'=>    fim := sim; exit;
        when others => null;
     end case;
   end loop;

put("enter output file name>");
   get_line(n1, l1); new_line;
   store_image(fim, n1(1..l), l1);
   put("image is stored"); new_line;
   loop
     put("Would you like to convert another ccd image? (y/n)>");
     get(answer2); skip_line;
     case answer2 is
         when 'n' | 'N' => exit MAIN_LOOP;
         when 'Y' | 'y'  => exit;
         when others =>null;
     end case;
    end loop;
```

```
  end loop MAIN_LOOP;
end CCD_to_256_usb_image;

-------------------------------------------------------------------
-------------------------------------------------------------------
--               Procedure FORMAT_256_IMAGE                      --
--                                                               --
-- Written by J. Thomas Walrond and Timothy Childress            --
-------------------------------------------------------------------
-------------------------------------------------------------------
--
--  A main program to convert 256 by 256 images in to the proper--
--  data type format.  Options to convert from the unsigned byte--
--  to the pseudo-signed byte format used in Marvin Hill's      --
--  Hologram_menu and the complex format of Kobel and Martin's  --
--  Executive.  The program also allows reformatting back to the--
--  USB format.                                                  --
-------------------------------------------------------------------

WITH INTEGER_TEXT_IO;                  USE INTEGER_TEXT_IO;
with DATA_TYPES;                         USE DATA_TYPES;
WITH IMAGE_TOOLS;                        USE IMAGE_TOOLS;
WITH TEXT_IO;                          USE TEXT_IO;
with DECLARATION_HANDLER;                use DECLARATION_HANDLER;
with FILE_IO;                            use FILE_IO;
with FILE_HANDLER;                       use FILE_HANDLER;
with DATA_STANDARD;                      use DATA_STANDARD;
with SSI_IO;                             use SSI_IO;

procedure FORMAT_256_IMAGE is

n2, n1          :  string(1..80);
answer1, l1,l2  :  integer;
answer2         :  character;
        im1             :  im_ar(1..256, 1..256);
        max_row, max_col:  integer;
        im2             :DATA_STANDARD.IMAGE_ARRAY_2D(1..256, 1..256);

begin
 MAIN_LOOP:
 loop
   --Make formatting options menu
```

```
new_line;
put("*******>> 256 by 256 IMAGE FORMATTING OPTIONS <<*******");
new_line; new_line; new_line;
put("   1.  Unsigned byte (USB) to Hologram_menu format.");
new_line;
put("   2.  USB to Executive (Kobel and Martin) format.");
new_line;
put("   3.  Hologram_menu to USB format.");
new_line;
put("   4.  Executive to USB format.");
new_line; new_line;
put("ENTER OPTION (1-4)> ");
get(answer1); skip_line;
--Get iput and output filenames
Put("ENTER FILENAME> ");
get_line(n1,l1);
new_line;

loop
   case answer1 is
      when 1|2    =>   get_image(n1(1..l1), l1, im1);exit;
      when 3      =>   SSI_IO.read_ssi_image(n1,l1,im2); exit;
      when 4      =>   FILE_HANDLER.GET_FILE(n1, l1, max_row, max_col);exit;
      when others => null;
   end case;
end loop;

put("Found file"); new_line;
put("ENTER OUTPUT FILENAME> ");
get_line(n2, l2);
new_line;
--format image
loop
  case answer1 is
    when 1 =>  USB_TO_HOLO_FORMAT(im1, n2, l2);exit;
    when 2 =>  USB_TO_EXECUTIVE_FORMAT(im1, n2, l2);exit;
    when 3 =>  HOLO_TO_USB_FORMAT(im2, n2, l2);exit;
    when 4 =>  EXECUTIVE_TO_USB_FORMAT(n1, l1, n2, l2, max_col, max_row);
               exit;
    when others => null;
  end case;
end loop;
```

```
   Put("Format another image? (y/n)> ");
   get(answer2); skip_line;

   loop
     case answer2 is
       when 'n' | 'N' => exit MAIN_LOOP;
       when 'Y' | 'y' => exit;
       when others => null;
     end case;
   end loop;

 end loop MAIN_LOOP;
end Format_256_image;

-------------------------------------------------------------------
-------------------------------------------------------------------
--                 PACKAGE Byte_io                               --
-------------------------------------------------------------------
-------------------------------------------------------------------

--   Declares a new input/output package "Byte_io
--   which is a instantiation of the sequential_io package with
--   the constraint that the data for input/output will be of the
--   type unsigned byte in a one dimensional array--the data
--   used to store the CCD camera images.
-------------------------------------------------------------------

with sequential_io;
with data_types; use data_types;

package byte_io is new sequential_io (element_type => unsigned_byte_ar_1d);
-- end Package Byte_io


-------------------------------------------------------------------
-------------------------------------------------------------------
--                    Package: Data_types                        --
-------------------------------------------------------------------
-------------------------------------------------------------------


--    Unsigned_byte_ar_1d is a one dimensional array
--    used to stored images.  im_ar is a two dimensional array of
--    intergers used to process images.
-------------------------------------------------------------------

with system; use system;

package data_types is

   type unsigned_byte_ar_1d is array (integer range <>) of unsigned_byte;
   type im_ar is array (integer range <>,integer range <>) of integer;

end data_types;


--------------------------------------------------------------
--------------------------------------------------------------
--                  Package:  File_io                       --
--------------------------------------------------------------
--------------------------------------------------------------

--   Package contains two procedures:
--   get_image and store_image.  These procedures enable
--   the input and conversion of unsigned byte format images
--   into integer format images for processing and display,
--   and the storage of integer format images in the unsigned
--   byte format (8 bit).
--------------------------------------------------------------
with system; use system;
with byte_io; use byte_io;
with data_types;        use data_types;

package file_io is

   procedure get_image (name1:  in string;
                        last:  in integer;
                        image:  out im_ar );

   procedure store_image(image:  in im_ar;
                         name1:  in string;
                         last:  in integer);
end;
```

```
package body file_io is

   procedure get_image (name1:in string;
                        last:  in integer;
                        image:  out im_ar ) is

      input:  byte_io.file_type;
      line:  unsigned_byte_ar_1d(image'range(2));

   begin

      open(file=>input,
           mode=>in_file,
           name=>name1(1..last));

      for r in image'range(1) loop
         read(input, line);

         for c in image'range(2) loop
            image(r,c) := integer(line(c));

         end loop;
      end loop;
      close(input);
   end;

-----------------------------------------------------------------------------
   procedure store_image(image:  in im_ar;
                         name1:  in string;
                         last:  in integer) is

      output:  byte_io.file_type;
      line:  unsigned_byte_ar_1d(image'range(2));

   begin

      create(file=>output,
             mode=>out_file,
             name=>name1(1..last));

      for r in image'range(1) loop
```

```
         for c in image'range(2) loop
            line(c) := unsigned_byte(image(r,c));

         end loop;
         write(output, line);
      end loop;
      close(output);
   end;
end File_io;

--------------------------------------------------------------------------
--------------------------------------------------------------------------
--                            IMAGE_TOOLS                               --
--                                                                      --
-- Written by J. Thomas Walrond and Timothy Childress                   --
--------------------------------------------------------------------------
--------------------------------------------------------------------------

--Description:  Tool package created to support the procedures
--CCD_to_256_USB_Format and Format_256_image. Individual description is
--listed with each procedure.

with data_types;            use data_types;
with file_io;     use file_io;
with text_io;     use text_io;
with float_text_io;     use float_text_io;
with DECLARATION_HANDLER;   use DECLARATION_HANDLER;
with DATA_STANDARD;         use DATA_STANDARD;
with SSI_IO;                use SSI_IO;
with FILE_HANDLER;          use FILE_HANDLER;

PACKAGE IMAGE_TOOLS IS

   procedure expand_ccd_image(im   :  in im_ar;
                              exim1: out im_ar);

   procedure BINARIZE_IMAGE(im   : in im_ar;
                            b_im: out im_ar);

   procedure CROP_CCD_IMAGE(im : in im_ar;
                            cim: out im_ar);
   procedure SAMPLE_IMAGE(im :  in im_ar;
                          sim:  out im_ar);

   procedure USB_TO_HOLO_FORMAT(im :  in im_ar;
                                n  :  in string;
                                l  :  in integer);

   procedure USB_TO_EXECUTIVE_FORMAT(im  :  in im_ar;
                                     n   :  in string;
                                     l   :  in integer);

   procedure HOLO_TO_USB_FORMAT(im  :  in DATA_STANDARD.image_array_2d;
                                 n  :  in string;
                                 l  :  in integer);

   procedure EXECUTIVE_TO_USB_FORMAT(n1       :  in string;
                                     l1       :  in natural;
                                     n2       :  in string;
                                     l2       :  in natural;
                                     Max_col  :  in integer;
                                     Max_row  :  in integer);


END;

PACKAGE BODY IMAGE_TOOLS IS
----------------------------------------------------------------------------
--                    Procedure EXPAND_CCD_IMAGE                          --
----------------------------------------------------------------------------

--  Takes ccd image (512 by 401 pixels) saved on the TARGA system and
--  expands the image to 512 by 512 by adding zeros evenly above and
--  below the image

   procedure expand_ccd_image(im:  in im_ar;
                              exim1: out im_ar) is

      exim: im_ar(1..512,1..512) := (others=>(others=>0));


   begin
```

```
      for r in im'range(1) loop
         for c in im'range(2) loop

            exim(r+55,c) := im(r,c);

         end loop;
      end loop;
      exim1 := exim;
   end;


-----------------------------------------------------------------------
--                  Procedure BINARIZE_IMAGE                         --
-----------------------------------------------------------------------

-- Thresholds images at pixel intensity value = 125 (intensity range 1-256).
-- If < 125 set pixel intensity to 0. If not, then set value to 250.

   procedure BINARIZE_IMAGE(im: in im_ar;
                            b_im: out im_ar) is

begin
  for r in im'range(1) loop
    for c in im'range(2) loop
      if im(r,c) < 125   then
         b_im(r,c) := 0;
      else
         b_im(r,c) := 250;
      end if;
    end loop;
  end loop;
end;


-------------------------------------------------------------------------
--                  Procedure CROP_CCD_IMAGE                           --
-------------------------------------------------------------------------
--  Grabs the center 256 by 256 section of an image grabbed by a TARGA
--  board using sony ccd camera.

   procedure CROP_CCD_IMAGE(im : in im_ar;
                            cim: out im_ar) is
```

```
begin

  for r in 73..328 loop
    for c in 129..384 loop
      cim(r-72, c-128) := im(r, c);
    end loop;
  end loop;
end;

-------------------------------------------------------------------------
--                     Procedure SAMPLE_512_TO_256_IMAGE               --
-------------------------------------------------------------------------

--Samples image by every other row and every other column to reduce image
--from 512 by 512 to 256 by 256.

   procedure SAMPLE_IMAGE(im :  in im_ar;
                                     sim:  out im_ar) is


begin

   FOR R IN 1..256 LOOP
      FOR C IN 1..256 LOOP
        SIM(R,C) := IM(2*R,2*C);
      END LOOP;
   END LOOP;
end;

---------------------------------------------------------------------------
--              Procedure  USB_TO_HOLO_FORMAT                            --
---------------------------------------------------------------------------

--  CONVERTS unsigned byte format images (already in integer form) to    --
--  the pseudo-signed byte format used in the Hologram_menu program      --
--  written by Marvin Hill                                               --

   procedure USB_TO_HOLO_FORMAT(im :  in im_ar;
                                n  :  in string;
                                l  :  in integer) is
```

```
     im2:  DATA_STANDARD.image_array_2d(im'range(1), im'range(2));

  begin

    im2:= DATA_STANDARD.image_array_2d(im);
    SAVE_SSI_IMAGE(n, 1, im2);
  end;

----------------------------------------------------------------------------
--                    Procedure USB_TO_EXECUTIVE_FORMAT                   --
----------------------------------------------------------------------------

--  Converts images in the unsigned byte format to the complex format used--
--  by Kobel and Martin's program Executive.                              --


  procedure USB_TO_EXECUTIVE_FORMAT(im  :  in im_ar;
                                    n   :  in string;
                                    l   :  in integer) is

     comp_im: DECLARATION_HANDLER.image_array_type(im'range(1), im'range(2));


  begin
    for r in im'range(1) loop
       for c in im'range(2) loop
          comp_im(r,c).imag := 0.0;
       end loop;
    end loop;
    for r in im'range(1) loop
       for c in im'range(2) loop
          comp_im(r,c).real := float(im(r,c));
       end loop;
    end loop;

    FILE_HANDLER.save_file(n, l, comp_im, "vid");

  end;


----------------------------------------------------------------------------
--               Procedure HOLO_TO_USB_FORMAT                             --
```

```
-----------------------------------------------------------------------------

-- Converts image in the HOLOGRAM_MENU image format (written by Marvin Hill)
-- to the unsigned byte format so the images can be displayed on the micro VAX
-- using Mike Roggemann's DISPLAY_IMAGE FORTRAN routine.

   procedure HOLO_TO_USB_FORMAT(im  :  in DATA_STANDARD.image_array_2d;
                                n   :  in string;
                                l   :  in integer) is

             usb_im: im_ar(im'range(1), im'range(2));

   begin

     for r in im'range(1) loop
       for c in im'range(2) loop
         usb_im(r,c) := im(r,c);
       end loop;
     end loop;
     STORE_IMAGE(usb_im, n(1..l), l);
    end;

---------------------------------------------------------------------------
--                     Procedure EXECUTIVE_TO_USB_FORMAT                 --
---------------------------------------------------------------------------

--  Convert image files created with Kobel and Martin's Executive to the --
--  unsigned byte format for display using Mike Roggemann's Display_image--

   procedure EXECUTIVE_TO_USB_FORMAT(n1       :  in string;
                                     l1       :  in natural;
                                     n2       :  in string;
                                     l2       :  in natural;
                                     Max_col  :  in integer;
                                     Max_row  :  in integer) is

          comp_im: image_array_type(1..max_row, 1..max_col);
               im: im_ar(1..max_col, 1..max_col);
          max_val: float;
            value: float;

   begin
```

```
     FILE_HANDLER.read_file(n1, l1, comp_im);

     --find maximum real value
     for r in comp_im'range(1) loop
        for c in comp_im'range(2) loop
            if max_val <= comp_im(r,c).real then
                max_val := comp_im(r,c).real;
            end if;
        end loop;
     end loop;

     -- normalize data
     for r in comp_im'range(1) loop
        for c in comp_im'range(2) loop
           value := (comp_im(r,c).real/max_val)*256.0;
           im(r,c) := integer(value);
        end loop;
     end loop;
     STORE_IMAGE(im, n2(1..l2), l2);
   end;

END Image_Tools;
```

Appendix E. *Code to Generate the ϕ Function*

This section presents an excerpt from the source code of Hologram_Menu written by Marvin Hill. This portion of the code was used to generate the phi, (Φ), function (described in Section 2.4) used to digitally model the optical $\ln r - \theta$, phase-only filter. The program adds the Φ term to a 256 by 256 pixel, real-valued image. Then Hologram_Menu Fourier transforms the resultant complex image using a Fast Fourier transform (FFT) routine written by Kobel and Martin.

The actual Φ function is dependent upon the wavelength of the light source, the focal length of the Fourier transforming lens, and the position of an image's pixels with respect to the image's center. These scaling factors do not exist for a FFT. Therefore, the Φ function was implemented with a general multiplicative scaling factor called "scale". Scale was adjusted for the most accurate looking coordinate transform of a 100 by 100 pixel square template. A scaling factor of 9.0 was used to generate the digital coordinate transforms for this thesis.

```
---------------------------------------------------------

---------------------------------------------------------

procedure ADD_LNR_THETA_PHASE( Work: in out
DATA_STANDARD.Complex_Array_R2d ) is

begin -- ADD_LNR_THETA_PHASE

--  Multiply by Ln r - Theta phase and take Fourier Transform

new_line;
XY_POS(1,Remarks_line);
put(" %REM - Multiplying by Ln r - Theta phase function...");
ERASE_LINE_FROM_CURSOR; new_line;

for X in Work'range(1)
loop
```

```
   for Y in Work'range(2)
   loop

      X1 := float( X - X_Center );
      Y1 := float( 257 - Y - Y_Center );

      if X1 /= 0.0 and Y1 /= 0.0 then
         Phi := 2.0 * pi * ( X1 * ( log( sqrt( X1**2 + Y1**2 )
)- 1.0 )- Y1 * atan2(Y1,X1) ) /SCALE;
      else
 Phi := 0.0;
      end if;

      In_Real := Work(X,Y).Real_N;
      In_Img  := Work(X,Y).Img_N;

      Work(X,Y).Real_N := In_Real * cos(Phi) - In_Img * sin(Phi);
      Work(X,Y).Img_N  := In_Real * sin(Phi) + In_Img * cos(Phi);

    end loop;
end loop;

end ADD_LNR_THETA_PHASE;
```

Bibliography

1. Casasent, David and Others. "Real-Time Deformation Invariant Optical Pattern Recognition Using Coordinate Transformations," *Applied Optics*, 26: 938–942 (March 1987).
2. Casasent, David and Demetri Psaltis. "Deformation Invariant, Space-Variant Optical Pattern Recognition," *Progress in Optics*, 16: 289–356 (1978).
3. Casasent, David and Demetri Psaltis. "New Optical Transforms for Pattern Recognition," *Proceedings of IEEE*, 65: 77–84 (January 1977).
4. Casasent, David and Demetri Psaltis. "Scale Invariant Optical Transform," *Optical Engineering*, 15: 258–261 (May 1976).
5. Fielding, Capt Kenneth H. *Position, Scale, and Rotation Invariant Holographic Associative Memory.* MS Thesis, AFIT/GEO/ENG/88D-2. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988.
6. Goodman, J. W. *Introduction to Fourier Optics.* New York: McGraw-Hill Book Company, 1986.
7. Horev, Moshe. *Picture Correlation Model for Automatic Machine Recognition.* MS thesis, AFIT/GE/EE/80D-25. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1980 (AD-A100 765).
8. Kabrisky, Dr. Mathew, Professor. Lecture notes taken in EENG 620, Pattern Recognition I. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1988.
9. Kobel, Capt William G. and Capt Timothy Martin. *Distortion Invariant Pattern Recognition in Non-Random Noise.* MS thesis, AFIT/GE/ENG/86D-20. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986 (AD-A177598).
10. Lee, Andrew J. and David P. Casasent. "Computer generated hologram recording using a laser printer," *Applied Optics*, 26: 136–138 (1 January 1987).
11. Martin, T. and Others. "A distortion invariant Pattern recognition algorithm," *Proceedings of SPIE*, 726: 63–72 (1986).
12. Mayo, 2nd Lt Michael W. *Computer Generated Holograms and Magneto-Optic Spatial Light Modulator for Optical Pattern Recognition.* MS thesis, AFIT/GEO/ENG/87D. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.
13. Miazza, D. and Others. "Optical In-Plane Distortion Invariant Pattern Recognition in Structured Noise," *Proceedings of the SPIE*, 939: 34–39 (April 1988).

14. Miazza, Capt David J. *Optical Pattern Recognition in Structured Noise.* PhD dissertation prospectus. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, January 1987.

15. *Operator's Manual HC-300 Holographic Recording Device.* Newport Corporation, Fountain Valley CA, undated.

16. Saito, Yoshiharu and Others. "Scale and Rotation Invariant Real Time Optical Correlator Using Computer Generated Hologram," *Optical Communications*, 47: 8–11 (August 1983).

17. *Sight-MOD(R) Development System Operations Manual.* Semetex Corporation, Torrance CA, 30 November 1987.

18. Troxel, 1st Lt Steve E. *Position, Scale, and Rotation Invariant Target Recognition Using Range Imagery.* MS Thesis, AFIT/GEO/ENG/87D-3. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

19. Wilson, Capt Jeffery A. *Optical Information Processing in a Confocal Fabry-Perot Resonator.* MS Thesis, AFIT/GEO/ENG/88D-5. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988.

Vita

Captain J. Thomas Walrond [redacted] He graduated from Cave Spring High School in Roanoke, Virginia in 1976 and attended Virginia Tech, where he earned the degree of Bachelor of Science in Biochemistry in 1980. He joined the United States Air Force and attended Officer Training School in May 1982 and was commissioned a Second Lieutenant in August 1982. An Air Force Institute of Technology lateral degree program sent Captain Walrond to the University of Missouri-Columbia in 1982 where he earned the degree of Bachelor of Science in Electrical Engineering in August 1984. Upon graduation, he was stationed at Scott Air Force Base, Illinois as a Navigational Aids Engineer, and later as Chief of Weather Radar Evaluation, with the 1866th Facility Checking Squadron. He entered the masters program in the School of Engineering, Air Force Institute of Technology in June 1987.

[redacted]

Vita

Captain Timothy G. Childress [redacted] 1975 [redacted] attended Southern Illinois University at Carbondale where he earned the degree of Bachelor of Arts in Biological Sciences. He delayed enlisted in the United States Air Force in September 1980 and came on active duty in January 1981. He accepted assignment to Officer Training School in September 1981 and was commissioned a Second Lieutenant in December 1981. An Air Force Institute of Technology lateral degree program sent Captain Childress to the University of Missouri-Columbia in 1982 where he earned the degree of Bachelor of Science in Electrical Engineering in December 1983. Upon graduation, he was stationed at Scott Air Force Base, Illinois as an Environmental Sensor Acquisition Project Officer with Headquarters Air Weather Service (MAC). He entered the masters program in the School of Engineering, Air Force Institute of Technology in June 1987.

[redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GE/ENG/88D-4

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL (If applicable)
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Air Force Institute of Technology (AU)
Wright-Patterson AFB, Ohio 45433-6583

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Rome Air Development Center

8b. OFFICE SYMBOL (If applicable)
RADC/COTC

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)
Griffiss AFB, NY 13441

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification) POSITION, SCALE, AND ROTATION INVARIANT OPTICAL PATTERN RECOGNITION FOR TARGET EXTRACTION AND IDENTIFICATION

12. PERSONAL AUTHOR(S) Timothy G. Childress, Captain, USAF
J. Thomas Walrond, Captain, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (Year, Month, Day)
1988 December

15. PAGE COUNT
148

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
17	05	
17	11	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Optical; Pattern Recognition; Correlation; Computer Generated Holograms; Position, Scale, and Rotation Invariance (PSRI).

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: POSITION, SCALE, AND ROTATION INVARIANT OPTICAL PATTERN RECOGNITION FOR TARGET EXTRACTION AND IDENTIFICATION

Thesis Chairman: Dr. Steven K. Rogers, Captain, USAF

JS Przemieniecki
10 Jan 89

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Dr. Steven K. Rogers, Captain, USAF

22b. TELEPHONE (Include Area Code)
513-255-6027

22c. OFFICE SYMBOL
AFIT/ENG

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

This thesis investigates the feasibility of optically implementing a real-time, pattern recognition system using correlation techniques in a position, scale, and rotation invariant (PSRI) feature space. Input target templates were optically Fourier transformed using an improved, high-resolution, high-pass filter positioned in the focal plane of the transforming lens. A logarithmic-polar coordinate transform of the magnitude-squared of the Fourier transform ($|FT|^2$) was performed with an improved method of focusing the magnified $|FT|^2$ onto a computer generated hologram (CGH), which was shown to scale linearly along the horizontal axis and logarithmically along the vertical axis. Optical, matched-filter correlations on the magnitude-squared Fourier transform logarithmic-polar (FLRT) feature space were performed using thermoplastic, phase-relief holography and Vander Lugt filtering.

Correlation results prove that scale and rotation changes of the input can be predicted accurately based on linear shifts of the correlation peak. Also, the FLRT feature space is shown to provide excellent discrimination for multiple-input scenes. The need for a cyclic correlation is verified, and digital simulations prove the validity of the optical results.